(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 487 871 B1

(12) EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention of the grant of the patent:
26.10.2005 Bulletin 2005/43

(21) Application number: **03710381.9**

(22) Date of filing: **17.03.2003**

(51) Int Cl.^{7}: **H04L 29/08**

(86) International application number:
PCT/JP2003/003140

(87) International publication number:
WO 2003/078459 (25.09.2003 Gazette 2003/39)

(54) METHOD, NETWORK AND APPARATUS FOR CONFIGURING AND CONTROLLING NETWORK RESOURCES IN CONTENT DELIVERY WITH DISTRIBUTED RULES

VERFAHREN, NETZWERK UND GERÄT ZUR KONFIGURATION UND STEUERUNG VON NETZRESSOURCEN BEIM ZURVERFÜGUNGSTELLEN VON INHALTEN MIT VERTEILUNGSREGELN

PROCEDE, RESEAU ET APPAREIL DE CONFIGURATION ET DE COMMANDE DES RESSOURCES D'UN RESEAU A REGLES REPARTIES DE FOURNITURE DE CONTENU.

(84) Designated Contracting States:
DE GB

(30) Priority: **18.03.2002 US 364585 P**

(43) Date of publication of application:
22.12.2004 Bulletin 2004/52

(73) Proprietor: **MATSUSHITA ELECTRIC INDUSTRIAL CO., LTD.**
Kadoma-shi, Osaka 571-8501 (JP)

(72) Inventors:
- **NG, Chan Wah**
  Singapore 271009 (SG)
- **TAN, Pek Yew**
  Singapore 547325 (SG)

(74) Representative: **Eisenführ, Speiser & Partner**
Patentanwälte Rechtsanwälte
Postfach 10 60 78
28060 Bremen (DE)

(56) References cited:
WO-A-01/77841 **US-A- 5 781 534**

- **SRISURESH P ET AL: "Middlebox communicatin architecture and framework;" INTERNET ENGINEERING TASK FORCE, XX, XX, 28 February 2002 (2002-02-28), pages 1-35, XP002211545**
- **BECK A, HOFMANN M: "IRML: A Rule Specification Language for Intermediate Services; Version 02" IETF INTERNET DRAFT, [Online] 21 November 2001 (2001-11-21), pages 1-27, XP002256751 Retrieved from the Internet: &lt;URL:www.globecom.net/ietf&gt; [retrieved on 2003-09-30]**
- **NG C W, TAN P Y, CHENG H: "Quality of Service Extension to IRML" IETF INTERNET DRAFT, [Online] July 2001 (2001-07), pages 1-13, XP002256752 Retrieved from the Internet: &lt;URL:www.globecom.net/ietf&gt; [retrieved on 2003-10-06]**
- **BARBIR A. ET AL: "Requirements for an OPES Service Personalization Callout Server" IETF INTERNET DRAFT, [Online] 7 March 2002 (2002-03-07), pages 1-25, XP002247308 Retrieved from the Internet: &lt;URL:www.globecom.net/ietf&gt; [retrieved on 2003-07-09]**
- **MA W-Y, SHEN B, BRASSIL J: "Content Services Network: The architecture and Protocols" PROCEEDINGS OF THE SIXTH INTERNATIONAL WORKSHOP ON WEB CACHING AND CONTENT DISTRIBUTION, [Online] 20 June 2001 (2001-06-20), pages 1-9, XP002256667 Boston, Massachusetts USA Retrieved from the Internet: &lt;URL:www.cs.bu.edu&gt; [retrieved on 2003-09-30]**

Printed by Jouve, 75001 PARIS (FR)

(Cont. next page)

- NG C W, TAN P Y, CHENG H: "Sub-System Extensions to IRML" IETF INTERNET DRAFT, [Online] June 2001 (2001-06), pages 1-8, XP002256753 Retrieved from the Internet: &lt;URL:www.globecom.net/ietf&gt; [retrieved on 2003-09-30]
- "INTERNET CONTENT ADAPTATION PROTOCOL (ICAP)" INTERNATIONAL CONFERENCE ON ANTENNAS AND PROPAGATION, XX, XX, 30 July 2001 (2001-07-30), pages 1-13, XP002226584
- SCHULTZRINNE H: "RTP: A transport protocol for real-time applications" NETWORK WORKING GROUP REQUEST FOR COMMENTS, XX, XX, 1 January 1996 (1996-01-01), XP002204956
- NG, C.W; TAN, P. Y.: "QoS and Delivery Context in Rule-Based Edge Services" 7TH INTERNATIONAL WORKSHOP ON WEB CONTENT CACHING AND DISTRIBUTION (WCW) , [Online] 14 August 2002 (2002-08-14), XP002256668 Boulder, Colorado, USA Retrieved from the Internet: &lt;URL:2002.iwcw.org&gt; [retrieved on 2003-09-30]

Description

Technical Field

[0001] The invention relates to the field of content delivery in a data communications network. More particularly, this invention pertains to the distributed control of data packets stream flowing through an intermediate network element, and the distributed control and configuration of network resources at a hierarchy of the intermediaries. The main intended use of this invention is to act as a proxy for content streaming, and providing content adaptation services in a distributed manner.

Background Art

[0002] Over the past decade, the Internet, or more specifically, the Internet Protocol (IP) based network, has seen a tremendous growth. The proliferation of the Internet and the increasing number of Internet users has resulted in extension and scaling problems for applications. This is especially true for applications designed for end-users, such as the World Wide Web (WWW), and audio-visual streaming. The increased in network bandwidth and processing power can hardly catch up with the demands of the increasing number of Internet users. This has resulted in longer load time for WWW page request, and lost of quality in real-time audio-visual playback across the Internet. The effort to reduce such undesirable effects has led to a wide deployment of intelligent network elements at the network edge (i.e. nearer to the end-users).

[0003] The most common use of such intermediate network elements are to function as caching proxies, such as hyper text transfer protocol (HTTP) proxies and/or caches as described in an article "Hypertext Transfer Protocol -- HTTP/1.1", IETF RFC 2616, June 1999, by Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee. These have been successful in reducing the network load at the WWW server and accelerating WWW contents delivery to the end-users. However, as the number of end-users increases, the variety of web-browser configurations and platforms widens. Similarly, the range of web contents is also broadening. Simply replicating static web contents cannot hope to sustain the ever-increasing demands from the end-users.

[0004] In addition, there has also been a noted increase in the deployment of multimedia streaming over the Internet. These usually employ the real-time streaming protocol (RTSP) as session protocol to set up and tear down communications channel, and real-time transport protocol and real-time control protocol (RTP/RTCP) for the actual transmission of content data. The RTSP is disclosed for example, in "Real Time Streaming Protocol (RTSP)", IETF RFC 2326, April 1998, by Schulzrinne, H., Rao, A., and Lanphier, R. The RTP/RTCP is disclosed for example, in "RTP: A Transport Protocol for Real-Time Applications", IETF RFC 1889, January 1996, by Schulzrinne, H., Casner, S., Frederick, R., and Jabcobson, V. Because of the versatile nature of Internet traffic, adaptation of the multimedia stream to the fluctuating traffic conditions is necessary to ensure a smooth presentation to the end-user. Though RTCP provides a means for end-users to report their communication status back to the sender, measures taken up by the sender based on receiver report can hardly be effective because the distance (network-sense) between the receiver and sender is often large. As most end-users connects to the Internet via an intermediary of some sort, for instance, firewall gateways, Network Address Translator (NAT), or proxies, the intermediary present a good choice to perform adaptation services on behalf of the content originator.

[0005] Furthermore, as the Internet grows, so does the range of devices that are used to access contents from the Web. This diversification of browser types has been accelerated with recent advancements in wireless Internet technology, whereby tiny handheld devices such as digital personal assistants (PDA) and mobile phones have microbrowsers built in that browse the web, or playback audio/visual streams. No longer can content authors develop contents with the assumption that the created content will only be viewed by users using traditional desktop computers. Device independence is now a critical consideration, as disclosed in "Device Independence Principles", W3C Working Draft, http://www.w3.org/TR/di-princ/, September 2001, by Gimson, R., et. al.,.

[0006] A number of international standardization organisations have recognized the need to provide services originally available only at the network core (where the servers are located) to the network edge (where the end-users are located). For instance, the Internet Engineering Task Force (IETF) has recently set up a few working groups focusing on providing services at the network edges. The Open Pluggable Extensible Services (OPES) working group is one such effort. The OPES working group focuses on extending the current HTTP proxies from performing simple caching task to a whole suite of adaptation services. The framework of OPES is specified in "A Model for Open Pluggable Edge Services", IETF Internet Draft, Work In Progress, http://www.ietf.org/internet-drafts/draft-tomlinson-opes-model-01, November 2001, by Tomlison, G., Chen, R., and Hofmann, M. There is also a Content Distribution Internetworking (CDI) working group that concentrates on the collaborations between different content distribution networks (CDN). Such collaboration efforts are believed to be able to further accelerate the delivery of contents to the end user.

[0007] The current use of intermediaries in content delivery is mostly restricted to providing simple functionality such as HTTP caching. HTTP proxy, or RTSP proxy. This cannot hope to maintain the service level demanded by the users. of today's Internet, as the number of end-users increases exponentially. Moreover, with the range of hardware devices and software agents employed to retrieve contents by different users are also broadening, content providers are finding it difficult to present to the users a coherent set of contents are that suited to the user's device and preferences.

[0008] Though various international bodies have recognized the above problems, and have acted to provide resolutions, their work could still be improved on. The OPES framework described in focused on the operations of a single intermediary, ignoring the current trend of collaborations between content delivery networks. In addition, though the idea of the OPES framework is to perform content adaptation so as to enhance the user experience in content retrieval, it focused only on parameters of the HTTP. This is not only inadequate for device independence, it also does not.cater to the growing number of audiovisual streaming applications.

[0009] Beck A. Hofmann, M.: "IRML: A Rule Specification Language for Intermediate Services; Version 02" IETF INTERNET DRAFT, [Online] 21 November 2001 (2001-11-21), pages 1-17 discloses web services as a new class of applications running on networked computers in a distributed environment. These services are invoked either directly by application end points or through intennediaries acting on behalf of application end points. Such intermediaries can appear in the form of caches, proxies, gateways, switches etc. and are also referred to as service dispatchers, application brokers, service brokers etc. IRML (intermediary Rule Mark-up Language) is designed to serve as a simple and efficient, but yet powerful language to express the service execution policies of application end points. IRML rules are typically processed by intermediaries that tricker the execution of web services according to these rules and policies.

[0010] Srisuresh P et al : « Middlebox communication architecture and framework; " INTERNET ENGINEERING TASK FORCE, 28 February 2002 (2202-02-28), pages 1-35 discloses that there are a variety of intermediate devices in the internet today that require application intelligence for their operation. Diagrams pertaining to real-time streaming applications such as SIP and H. 323 and peer-to-peer application such as Napster and NetMeeting can not be identified by nearly examining packet headers.

Disclosure of Invention

[0011] To solve the problem listed in section 3.3, the present invention allows content providers, access providers, and/or end-users to specify rules governing the delivery of content via intermediate network elements. These rules can be distributed to other intermediaries along the content flow path, to achieve the maximum efficiency and easier control of network resources. It is suitable for deployment by different content delivery networks, and can cooperate among one another. In addition, the current invention allows rules to be specified that are specially catered for real time content streaming. The present invention also defines a mechanism to extend rules to be construct based on user preferences, and device capabilities. Such a provision allows the rule author to construct rules that can better adapt contents to achieve device independence.

[0012] This invention is defined by the appended claims and involves the operations of one or more intermediate network elements performing content delivery and adaptation between end-users and the content providers. The intermediate network element (also known as intermediary) will parse each data packets transferred between the end-user and the contents provider. When the data packets matches certain criteria as specified by a set of rules registered with the intermediary, actions specified in the rules are carried out, usually results in the modification of the data packets. Rules in an intermediary can be distributed to other intermediaries which are more suited to evaluate the rule and/or perform the adaptations. In addition, rules can also cater specially to real time streaming protocol, or be constructed with delivery context parameters to achieve device independence.

Brief Description of Drawings

[0013]

Fig. 1 is a framework of an Intermediate Network Element, showing the functional architecture of the intermediate network element as used in the invention.

Fig. 2 shows nodes along the Content Path, and illustrates a typical content flow path from the content server to the content user, traversing a single or plural number of intermediaries.

Fig. 3 shows example of ContentPath Structure, particularly showing the values stored in a ContentPath structure of the intermediary foo4.bar.com as marked by literal 204 in Fig. 2.

Fig. 4 shows a method of Extracting Intermediaries Information from Embedded Signature in Data Packets, particularly showing the flow diagram of the method to extract intermediaries' signatures in the data packets to construct/update the ContentPath structure.

Fig. 5 shows a method of Determining the Remote Intermediary to Distribute Rule, particularly showing the algorithm used to determine the remote intermediary to distribute a rule to, given the distribution indication.

Fig. 6 shows a method of Parsing Rule with Distributed Rule Support, particularly showing the algorithm used to parse a rule with the focus on supporting distributed rules. The actually method of parsing the rule to check for syntactical validity and evaluation of the rule is outside the scope of this document.

Fig. 7 shows a method of Determining the Remote Intermediary to Distribute Rule in a Server-Client Model, particularly showing the algorithm used to determine the remote intermediary to distribute a rule to, given the distribution indication, in a server-client model.

Best Mode for Carrying Out the Invention

[0014] An apparatus and methods for distributed network resource management is disclosed. To facilitate understanding of the invention, the following definitions are used:

[0015] A "packet" is a self-contained unit of data of any possible format that could be delivered on a data network.

[0016] An "intermediary" and an "intermediate network element" are equivalent, and are used interchangeably, unless otherwise specified, to refer to a gateway, a router or an intelligent network hub for which this invention applies to.

[0017] The term "current intermediary" or "current intermediate network element" refers to an intermediate network element that is processing a data packet, or a rule specification, depending on the context the term is used in.

[0018] The terms "content server" and "content user" are used with respect to a sever-client model of information exchange. The content user, which is the client, will send a single or plural number of data packets to the content server containing a request. Such data packets are known as request packets. The content server upon processing the request would reply with a single or plural number of data packets containing the response. Such data packets are referred to as response packets.

[0019] In distribution of rules, the term "target intermediary" or "target intermediate network element" refers to the intermediate network element of the current invention receiving the distributed rule. The term "distributing intermediary" or "distributing intermediate network element" refers to the intermediate network element of the current invention distributing the rules to other intermediaries.

[0020] In the following description, for purpose of explanation, specific numbers, times, structures, and other parameters are set forth in order to provide a thorough understanding of the present invention. However, it will be apparent to anyone skilled in the art that the present invention may be practiced without these specific details.

[0021] The intermediate network element for which the current invention applies to, consists of the functional architecture as depicted in Fig. 1. The intermediary consists of a gateway module (101), a rule engine (102), a single or plural number of special packages (103), and the rule injection module (104).

[0022] The gateway module (101) is the collection of functional blocks that implement gateway or proxy functionalities. These can include, but not limited to, HTTP proxies and/or caches, RTSP proxies, RTP/RTCP mixers and/or translators, and application level gateways (ALG). For instance, we consider an intermediary performing the role of a HTTP proxy. The gateway module (101) would thus implements the functional component to handle HTTP connections from the client side, the functional components to establish a HTTP connection to the server, or another HTTP proxy, based on the client side request, and the functional component to relay responses from the server back to the client side. Effectively, the gateway module (101) is the functional component that implements the protocols (eg HTTP, RTSP) for which the intermediary is an active party of.

[0023] The rule engine module (102) parses through all or part of the data packets that passes through the intermediate network element and matches these data packets to a set of criteria specified by a singular or plurality of rules. This is known as evaluation of rules. These rules are specified in a logical unit known as a Rule Specification. A Rule Specification can contain a singular or plural number of rules. When a match is found, the corresponding action(s) specified in the rules is(are) triggered. This is known as the "firing of a rule". The action performed can include, but not limited to, inserting contents to the data packets, removing part or all of the contents from the data packets, and modifying contents in the data packets. These insertion, removal, and modification of packets contents can be carried out on the intermediary, or some other remote machines dedicated to perform packets transformations.

[0024] Examples of rules that are parsed by the Rule Engine (102) include rule which determines the bandwidth to be allocated to the data stream that the client is requesting. For instance, a rule may be specified in the following high-level form:

"If network channel of client <=1 Mbps, then allocate 10 kbits for data stream". (Rule 1)

[0025] The rule engine module (102) implements the functionality to parse such rules and determines if a match (i. e. if the end-user's network channel has a capacity is less than or equals to 1 Mbps) occurs. The above example also shows how the rules control the network resource allocation decisions.

[0026] Another example of rules being parsed by the rule engine module (102) will be to determine the next intermediary or server to contact in response to a client's request received from the gateway module (101). Here, base on the parameters of the request, the request may be routed to a different server/intermediary. For instance, consider the following high-level rules:

"If requested data is audio only, route request to foo3.bar.com" (Rule 2a)
"If requested data is audio and video, route request to foo4.bar.com" (Rule 2b)

[0027] The rule engine module (102) is responsible in parsing and interpreting such rules, and determining if one of condition is met. When one is met, the rule engine module (102) will inform the gateway module (101) which next intermediary/server is selected, and the gateway module (101), which implements the actual functionality to communicate with others using the specific protocol, would proceed to route the request to the selected next intermediary/server.

[0028] The intermediary in the current invention can have zero, single or plural number of special packages (103). These special packages (103) are modules designed to enhance the rule engine module (102) by providing specialized functionalities. For example, a Quality-of-Service (QoS) special package can be employed to assist the rule engine module (102) in understanding and evaluating rules that involves QoS parameters and conditions. The rule engine module (102) on its own can only parse rules and try to find a match on the conditions spelt out by the rule specifications. Using the example of (Rule 1a) above, the rule engine module (102) will need help to determine the actual capacity of the network channel of the client. A special package (103) for evaluating QoS parameters can be installed in the intermediary to evaluate such expression. The rule engine module (102) would query the QoS special package (103) when it parses a rule specification that specifies a QoS parameter (such as bandwidth, delay, etc). The QoS special package (103) evaluates the value of the parameter in question and passed it back to the rule engine module (102). From there, the rule engine module (102) can then proceed to check if a match has occurred in the condition specified by the rule specification. In this way, a modular design of the intermediary can be achieved. The rule engine module (102) performs the job of parsing rule specifications. It utilizes different special packages (103) to evaluate the value of a parameter that is specified in a rule specification, and from the values, determines if a condition is matched. In the current invention, a special package for evaluating rules based on delivery context is set forth.

[0029] The rule injection module (104) is a module that dynamically loads and unloads rules to and from the rule engine module (102). It also provides the interface for remote parties to dynamically register, activate or de-activate rules. This module is indispensable in supporting distribution of rules through various intermediaries.

[0030] Initially, when the intermediary starts up, the rule injection module (104) will load an initial set of rule specifications, based on a configuration file or otherwise, from the storage of the intermediary. The rules specifications will be loaded to the rule engine module (102). When a client request (or server response data) arrives, the gateway module will processing the clients request (or server response data), and pass it to the rule engine module (102) for rule parsing. The rule engine module (103) will parse the rule specifications and try to seek for a match in the conditions specified against the request (or response). While parsing these rules, the rule engine module (102) may require assistant from special packages (103) to evaluate the value of parameters.

[0031] The rule injection module (104) also allows rules to be dynamically loaded to the intermediary. For example, an administrator may remotely transfer a new set of rule specifications to be installed on the intermediary. Alternatively, the administrator may want to remotely remove a rule specification from the intermediary. The rule injection module (104) handles such remote operations. In addition, these operations need not be limited to human administrators. Indeed, the first portion of the current invention details the mechanism that allows rule specifications to be loaded and unloaded dynamically between intermediaries. The rule injection module (104) plays the role here of accepting connections from other intermediaries and handle requests to load or unload a rule specification to/from the rule engine module (102).

[0032] Distribution of rules implies that rule specification(s) that is(are) loaded on one intermediary can be passed to another intermediary to be evaluated. Whole or part of the rule specification may be indicated to be distributed. These indications also suggested to which intermediary along the data flow path to distribute to. Fig. 2 shows a typical content flow path. Note that there may be other network elements performing relaying task along the content path that are not shown in Fig. 2. Along the path from the content server (201) to the content user (207), there may be a single or plural number of intermediaries (202 - 206) with the current invention employed.

[0033] Authors of the rule specification can indicate which intermediary along the content flow path to distribute the rule specification partially or wholly. Since it is unreasonable for authors to know how intermediaries are deployed in an actual real world situation, authors specify the preferred intermediary where the rule is distributed to by indicating the direction of the distribution, i.e. towards the source node or towards the destination node. The term "source" and "destination" are used with respect to the data packets. The source node is the node that generates the data packet, and the destination node is the node that consumes the data packet. In a server-client model, the directions may be specified as "towards server" or "towards client".

[0034] For example, using the deployment scenario as illustrated in Fig. 2, a rule specification is submitted to the intermediate network element foo4.bar.com (204). A part of the rule is indicated to be distributed towards the "destination". When processing a request packet, i.e. packet sent from the content user (207) to the content server (201), this portion of the rule specification can be distributed to the intermediary foo2.bar.com (202) or the intermediary foo3.bar.com (203). Conversely, when processing a response packet, the same portion of the rule specification can be distributed to the intermediate network element foo6.bar.com (206) or the network element foo5.bar.com (205). Similarly, when a part of the rule is indicated to be distributed towards the "server", it can be distributed to the intermediary foo2.bar.com (202) or the intermediary foo3.bar.com (203). Conversely, when a portion of the rule specification is indicated to be distributed towards the "client", that portion of the rule specification can be distributed to the intermediate network element foo6.bar.com (206) or the network element foo5.bar.com (205).

[0035] One objective of the current invention is to allow rule authors to be able to specify a rule hierarchy where the topmost level of the rule specification resides on one intermediary and the lower level portions of the rule specification reside on other intermediaries. This allows efficient control of the intermediary operations. Thus, in addition to specifying the direction at which distributed rules should flow (i.e. forward, backward, towards content server or towards content user), the current invention also allows rule authors to specify the approximate location in that direction where the rule should be distributed.

[0036] Using the above example, a rule author may specify the distributed portion of the rule specification to be distributed to the intermediary as near to the content user as possible. In Fig. 2, this implies the intermediary foo6.bar.com (206). Alternatively, a portion of the rule might be specified to be distributed to the intermediary one hop away from the element nearest to the content user. In Fig. 2, this implies the intermediate network element foo5.bar.com (205). Similarly, it is possible for the rule author to specify that a portion of the rule to be distributed to the next intermediary towards the content server. Using the previous example where the rule specification is submitted to foo4.bar.com (204), this means the rule author wanted the portion of the rule to be distributed to the network element foo3.bar.com (203).

[0037] The current invention covers all the afore-mentioned means of marking the rule specification for distribution. As an illustration, the following character-based indication methods are presented. It should be apparent to anyone skilled in the art that other forms of indications can be used to achieve equal effect, such as using numeric or alphanumeric codes. In the character-based indications, each indications is of the form
<target direction>-<approximate location from target> or of the form
<approximate location towards target>-<target direction>
where <target-direction> is the term source, destination, server or client, and <approximate location from target> and <approximate location towards target> are numerical values indicating the number of intermediaries away from the specified target.

[0038] For instance, to indicate the portion of rule to be distributed to the intermediary nearest to the content server, the rule author may use an indication of server-1 to show that the rule should be distributed to the intermediary that is 1 hop away from the content server. When the indication of 2-server is used, the rule author expressed the desire to distribute the rule to an intermediary that is 2 hops away from the intermediary where the rule is loaded, towards the direction of the content server. Similarly, the indication client-2 indicates the rule should be distributed to the intermediate network element that is 2 hops away from the content user, and the indication 1-client indicates that the rule should be distributed to the next intermediary in the direction of the content user.

[0039] In order for intermediaries to distribute rule specifications among themselves, the intermediaries must have a way to first discover the existence of other intermediaries on a given content flow path. Each intermediary may also be connected to multiple content servers, content users and other intermediaries, thus discovery may not be feasible to be performed statically using configuration files, nor one-shot at system starts up.

[0040] The present invention requires that intermediary to embed an indication of their presence in the content as data packets flows from the content user to the content server and vice versa. Such an indication is known as the signature of the intermediary. These signatures should contain information of the intermediary, such as the resolvable hostname and the capabilities of the intermediary. Capability of the intermediaries should clearly indicate that the intermediary support distributed rules, and should also include information such as the special packages installed in the intermediary.

[0041] For example, in the HTTP and RTSP protocols, intermediaries can append their signature to the "Via" general header found in the request and response headers. An intermediary of hostname foo4.bar.com with installed QoS special package can insert the following "Via" header field as its signature:

```
Via: 1.1 foo4.bar.com (OPES=standard,qos,distributed)
```

[0042] For other protocols which do not have built-in mechanism for intermediaries to embed their signature, other means could be sought for. For instance, protocols usually provide the functionality for machines to embed optional information into the data packets (normally using optional extension headers). This can be used to carry signatures of the intermediary. In addition, the above example used character strings as the signature for ease of understanding. It should be apparent to anyone skilled in the art that other forms of signature can be used to achieve equal effect, such as using numeric or alphanumeric codes, so long as an external entity can extract the hostname and capabilities of the intermediary from the signature.

[0043] In both cases where the protocol built-in mechanism is used, or optional extension is used, multiple signatures should be allowed so that signature of each subsequent intermediary can be appended. In other words, when a data packet reach any given intermediate network element in the content flow path, the intermediate network element knows the other intermediaries the data packet has previously traversed. This also enables the intermediary to know the order of the intermediaries the packet traversed.

[0044] In a typical operation, there will be request flowing from the content user to the content server, and the response flowing from the content server to the content user. Intermediaries will thus know all other intermediaries along the content flow path once a pair of request and response data packets passed through them. For instance, using the scenario shown in Fig. 2, the intermediary foo4.bar.com (204) will know the existence of the intermediaries foo6.bar.com (206) and foo5.bar.com (205) when the request from the content user (207) reaches it. When the content response from the content server reaches foo4.bar.com (204), the intermediary will discover the existence of the intermediate network elements foo2.bar.com (202) and foo3.bar.com (203). Thus, the intermediary foo4.bar.com (204) will be able to detect the presence of other intermediate network elements in the entire content flow path.

[0045] Intermediaries will maintain a cache of known intermediaries network along any given content flow path. The reason for doing so is explained below. When an intermediary received a request, it may be necessary for it to distribute a rule to another intermediate network element towards the content server. If the intermediary relies only on the embedded signature to discover other intermediate network elements, it cannot know in advance other intermediaries in the forward path towards the content server, until it receives the content response. Should such a situation arise, the rule engine module (102) should check if it could retrieve information of intermediaries from its cache. If it can, then the rule can be distributed to a forward intermediary; else, the rule should be evaluated locally.

[0046] To maintain the cache, the data formats as shown in Data Format 1 and Data Format 2 below can be used. Data Format 1 is used to record the host identification (stored in the field hostname) and capabilities (stored in the field capabilities) of the intermediary. Data Format 2 is used to record the list of known intermediaries along a given content flow path. The content flow path is uniquely specified by the source (stored in the field source), destination (stored in the field destination), and protocol (stored in the field protocol) triplet, expressed as {source, destination, protocol}. The num_nodes field store the number of intermediate network element that a data packet from the source node must traverse before reaching the destination node starting from (but excluding) the current intermediary, and the nodes array store the information of each of such intermediate network element. Fig. 3 illustrates a pair of ContentPath data formats stored in the intermediary foo4.bar.com (204) in the scenario depicted Fig. 2.

```
struct IntermediaryEntry {
      NodeID    hostname;
      char      capabilities[];
}
```

Data Format 1: Intermediary Entry

```
struct ContentPath {
      NodeID                          source;
      NodeID                          destination;
     ProtocolType                    protocol;
     int                             num_nodes;
     struct IntermediaryEntry        nodes[];
}
```

Data Format 2: Data Flow Path Information

[0047] Fig. 4 depicts the method devised to extract the intermediaries' signatures embedded in the data packets and construct/update the ContentPath structure. When a data packet arrives, the intermediary first check if there is any signature embedded, as shown in the step marked with literal 401. If there is one, a ContentPath structure is searched from the cache that matches the {destination, source, protocol} triplet, as shown in step marked with literal 402. Note that the source of the data packet is checked for against ContentPath.destination, and vice versa. The reason for this is because the ContentPath structure is used to give the list of intermediaries towards the destination node, whereas when extracting signatures from the data packets, the intermediaries are given from the source node. Thus, there is a need to swap the destination and source nodes when searching for a match.

[0048] If none can be found, a new ContentPath structure is allocated, as shown in step marked with literal 403. When a cached ContentPath structure is located, the num_nodes and nodes fields will be purged, as shown in the step marked with literal 404. A last-in-first-out stack to store the signatures temporarily is then initialised to be empty and the counter n is set to zero in the step marked with literal 405. In the step marked with literal 406, each embedded signature is extracted and pushed to the stack. In addition, the counter n is incremented to record the number of signatures extracted. When all signatures are extracted, the counter n will contain the number of intermediaries before the current network element in the content flow path. This is stored to the num_nodes field. Signatures in the stack are then popped out to update the nodes array, as shown in the step marked with literal 406.

[0049] Previous description has presented the method for intermediaries to discover other intermediaries along the content flow path. When a rule engine module parses a rule specification and found that a portion of the rule specification is marked to be distributed, it can then check the appropriate ContentPath structure (by using the content server, content user, and protocol triplet) and determine which remote intermediaries to distribute the rule to.

[0050] Fig. 5 shows the algorithm used to determine the remote intermediary to distribute the rule to, given the indication of distribution in the form of
<target direction>-<approximate location from target>
or
<approximate location towards target>-<target direction>
as previously described. The term target is used to denote the numerical value of <approximate location towards target> or <approximate location from target>. The term directive contains the value "from" if the first form is used, and the value "to" if the second form is used. The term direction is the value "source" or the value "destination". The term src, dst and protocol refers to the source, destination and protocol extracted from the data packet respectively.

[0051] The algorithm first searches for the ContentPath data format as shown in the steps marked with 501 through 504. If the direction of distribution is towards the destination, the triplet {src, dst protocol} is used to locate the ContentPath, as shown in the step marked with literal 502. Else, the triplet {dst, src, protocol} is used instead, as shown in the step marked with literal 503. If no ContentPath can be found, the algorithm returns NULL, as shown in the step marked with literal 512. In the steps marked with 505 and 506, target in checked to prevent it from exceeding the number of remote intermediaries in the direction of distribution. In the step marked with literal 507, the distribution indication is checked to see if it is in the form
<target direction>-<approximate location from target>
or
<approximate location towards target>-<target direction>

[0052] For the first form, the numerical value indicates the number of intermediary from the end host (content server or content user). However, the intermediaries are listed in nodes array in the order of the direction towards the end node. Thus, in the step marked with literal 508, a temporary variable x is set to the number of intermediaries minus the numerical value target. Conversely, if the indication is in the second form, then the temporary variable x is set to the numerical value target minus 1, as shown in the step marked with literal 509. The reason to subtract one is because the first element in the nodes array is assumed to be nodes[0]. Anyone skilled in the art can easily modify the above formulae to suit other kinds of array arrangement. In the step marked with literal 510, the variable x is checked to see if falls out of range. If it does, the algorithm returns NULL to indicate no suitable remote intermediary can be found, as shown in the steps marked with literal 512. Else, the function returns the remote intermediary given in nodes[x], as shown in the step marked with literal 511.

[0053] Fig. 6 shows the method of parsing a rule specification with the consideration of distributing rules. The rule specification is first parsed to check for syntactical validity, and invalid rules are rejected, as shown in the steps marked with 601, 602, and 603. The rule is next checked to see if the it is marked to be distributed, as shown in step marked with literal 604. If it is not marked, the rule is evaluated locally (605). Else, the algorithm depicted in Fig. 4 is used to identify the remote intermediary to distribute the rule to, as shown in the step marked with literal 606. If the algorithm returns NULL, that means no suitable remote intermediary can be found, then the rule is evaluated locally, as shown in steps marked with 607 and 605. When a remote intermediary is found, it is checked to see if it supports the special package required by the rule, as shown in step marked with literal 608. If it does not, the rule is evaluated locally (605). If it does, the rule is then distributed (609). The whole process is repeated for the next rule to be parsed (610).

[0054] If a server-client model is used, where the target direction can be specified by towards "server" or towards "client" instead, then the method of locating the target intermediary given in Fig. 5 can no longer be used. Fig. 7 shows the method for a server-client model. The only difference between the algorithm in Fig. 7 and the one in Fig. 5 is in the steps marked with literals 701 through 703, and the steps marked with literals 501 through 503. In the step marked with literal 701, the target direction is first checked if it is towards server or client. If the target direction is server, the ContentPath is searched using the { node identification of client, node identification of server, protocol } triplet, as shown in the step marked with literal 702. If the target direction is client, the ContentPath is searched using the { node identification of server, node identification of client, protocol } triplet, as shown in the step marked with literal 703. The remaining steps marked with literals 704 through 712 are identical to the steps marked with literal 504 through 512.

[0055] In order to distribute the rule, the intermediary needs to signal the receiving intermediary. For ease of explanation, the scenario that is illustrated in Fig. 2 is used. For the following discussion, the intermediary foo4.bar.com (204) is the network element that loads the rule, and it has determined that the rule needs to be distributed to the intermediary foo6.bar.com (206) for evaluation.

[0056] To signal foo6.bar.com (206), foo4.bar.com (204) can embed a signal into the data packet. The present invention requires that the embedded signal contain the identifier of the intermediary that is distributing the rule, the identifier of the intended intermediary receiving the rule, and the rule identifier that uniquely identifies the rule to be distributed. The rule identifier must uniquely identify the portion of the rule specification on a given intermediary that is to be distributed, and the identifier should not vary with time.

[0057] For example, in the HTTP and RTSP protocols, intermediaries can append tokens to the "Pragma" general header in the request and response headers. Thus, the current invention can make use of this to embed the required signals. For example, foo4.bar.com can append the following token OPES-distributed="foo6.bar.com:XYZA-BC@foo4.bar.com"; to the "Pragma" general header of the response. The token used is of the form
OPES-distributed="<target>:<rule identifier>@<distributor>"
where <target> refers the hostname of the intermediary to receive the distributed rule, <rule identifier> refers to the unique identifier to identify the distributed rule, and <distributor> refers to the intermediary that is distributing the rule.

[0058] For other protocols which do not have built-in mechanism for intermediaries to embed signals, other means could be sought for. For instance, protocols usually provide the functionality for machines to embed optional information into the data packets (normally using optional extension headers). This can be used to carry signals for intermediary. In addition, the above example used character strings as the signature for ease of understanding. It should be apparent to anyone skilled in the art that other forms of signature can be used to achieve equal effect, such as using numeric or alphanumeric codes, so long as an external entity can extract the hostname of the distributing intermediaries, hostname of the target intermediary and the rule identifier from the signal.

[0059] In both cases where the protocol built-in mechanism is used, or optional extension is used, multiple signals should be allowed so that two or more sets of rules can be distributed at a single passing of a data packet. Every intermediary must inspect the data packets to detect such signals, and check if the signal is intended for it. Once it determined the signal is intended for it, the intermediary can optionally remove the signal from the data packet.

[0060] The rule identifier is used to retrieve the actual rule from the distributing intermediary using a separate communications channel. The rule injection module (104) is responsible for establishing such a communications channel and retrieving/passing any distributed rule. The current invention does not specify the format of such a communications channel. Because the rule identifier is unique given a specified intermediary, intermediaries should cache the retrieved distributed rule using the rule identifier and the hostname of the distributing intermediary as a cache key. This eliminates the need to retrieve the same distributed rule should a subsequent distribution occur again.

[0061] The above mechanisms can be deployed for contents distribution where there is a plurality of sending and receiving nodes. Such situation is decomposed into multiple data flow paths, each containing one sending node and one receiving node. Note that this decomposition is only used for the construction of the ContentPath structure as described previously. When an actual data that arrives at an intermediary that is sent to plurality of receiving nodes, the intermediary can then decide on the rule distribution based on each ContentPath structure for each corresponding sending node (i.e. source) and receiving node (i.e. destination) pair. If the targeted intermediary to distribute a rule to happens to be the same, then a single signal can be embedded onto the content. If more than one target intermediary is identified (because the content path split somewhere along the line), one separate signal for each targeted intermediary can be embedded into the content.

[0062] In the previous descriptions, the intermediate network for distribution of rule is revealed. This document will, in the following discussions, turn to the next portion of the current invention, which narrows the deployment of the current invention to real time content streaming situation. In this situation, the content user sends a request to the content server, via a single or plural number of intermediary(intermediaries) which is(are) the object(s) of the current invention, to set up a real time session. When the content server accepts the request with an appropriate response, a communications channel is set up between the content server and the content user through the intermediary(intermediaries). This communications channel between the content server and content user is hereafter referred to as the content session. The content server starts transmitting data packets through the content session to the content user without any active request from the content user, until the content user sends a request, via the intermediary, to tear down the content session. Such data packets sent spontaneously by the content server are henceforth referred to as content packets. During the course of the transmission of content packets by the content server, the content user may or may not transmit information about the transmission statistics back to the content server. Such statistics are hereafter referred to as feedback packets.

[0063] One existing protocol that fits the above description is the Real Time Streaming Protocol (RTSP). However, the current invention can be applied to other protocols that exhibit the same behaviour previously described, as should be apparent to anyone skilled in the art.

[0064] For all known prior arts, rules are evaluated for each request and/or response packets that passes through the intermediary. The current invention extends this by providing the capability for rule authors to specify rules that are evaluated whenever a content packet passes through the intermediary, rules that are evaluated whenever a specifies multiple of content packets passes through the intermediary, rules that are evaluated when a feedback packet passes through the intermediary, and rules that are evaluated at a specified regular interval throughout the duration when the content session is established. To attain such provision, rule authors are allowed to tag each rule with a special attribute.

For purpose of explanation, the attribute is referred to as the "evaluateOn" attribute. Table 1 below listed the possible values of the "evaluateOn" attributes. Anyone skilled in the art should recognized that the current invention can be deployed using other names.

Table 1:

"evaluateOn" Attributes	
"evaluateOn" Attribute	Description
"request"	the rule is to be evaluated upon the reception by the intermediary of a request packet from the content user to the content server
"response"	the rule is to be evaluated upon the reception by the intermediary of a response packet from the content server to the content user
"content"	the rule is to be evaluated upon the reception by the intermediary of a content packet from the content server to the content user
"feedback"	the rule is to be evaluated upon the reception by the intermediary of a feedback packet from the content user to the content server
"x-contents"	the rule is to be evaluated upon the reception by the intermediary of x multiple number of content packets from the content server to the content user, where x is a specified numerical value
"t-seconds"	the rule is to be evaluated when a content packet is received upon elapsed of every t seconds interval for as long as the content session is established, where t is a specified numerical value

[0065] The last part of the current invention concerns the employment of special packages (103). As described earlier, special packages (103) are modules that enable the rule engine module (102) to evaluate rules involving different set of parameters (such as Quality of Service). The current invention defines a new special package, known as the "Delivery Context" special package. This package will allow the rule engine module (102) to interpret rules that are constructed based on delivery context. Four major classes of delivery context are defined, as shown in Table 2 below. These are User Preferences, Agent Capabilities, Device Capabilities, and Natural Environment. User Preferences refers to information about the human user, including browsing preference, language preferences, display preferences, QoS preferences, age group and gender. Agent Capabilities provide information on the software agent, such as the agent type, supported formats, supported languages, and supported transport protocols. Device Capabilities refers to the information about the hardware device, which include the device type, processor speed and type, memory capacity, screen resolution and depth, and operating systems. Natural Environment provide information about the natural environment surrounding the end user, including whether the end user is indoor or outdoor, the end user's velocity, location of the end user, and illumination properties.

Table 2:

Delivery Context Parameters	
User Preferences	
Parameters	Values
"browsing-preference"	descriptive text about the user's browsing preferences, such as text only, image sizes, handicaps accessibilities options, searching and filtering preferences
"language-preference"	descriptive text about user's order of language preferences
"display-preference"	descriptive text about user's color preferences, full screen or window
"age-group"	descriptive text about the user's age group
"gender"	descriptive text about the user's gender
"employment"	descriptive text about the user's job nature
Agent Capabilities	
Parameters	Values

Table 2: (continued)

Delivery Context Parameters	
User Preferences	
Agent Capabilities	
"agent-type"	descriptive text about the software agent
"supported-formats"	descriptive text about the content formats supported, and content encoding supports
"supported-languages"	descriptive text about the language supported by the software agent
"supported-protocols"	descriptive text about the transmission protocols supported by the software agent, and whether it has multicasting, broadcasting capabilities
Device Capabilities	
Parameters	Values
"device-type"	descriptive text about the device type
"processor"	descriptive text about processor speed and family
"memory-capacity"	descriptive text about memory capacity of the physical and secondary memory
"screen"	descriptive text about resolution and depth
"operating-system"	descriptive text about the operating system type
Natural Environment	
Parameters	Values
"location"	descriptive text about the user's location, such as indoor or outdoor, and the locale
"mobility"	descriptive text about the user's mobility, whether fixed or moving, and the velocity if moving
"illuminations"	descriptive text about illuminations surrounding the end user

[0066] The Delivery Context special package interprets rules that are constructed using parameters that are from delivery context. There are various methods where the delivery context special package can obtain the actual values of the parameters. One method is to establish a communications channel with an external entity that provides knowledge of such parameters, for example the content user. For parameters such as the Device Capabilities, it might be necessary to obtain it from the content user directly. An alternative method is to obtain it from another module that resides on the intermediary. This module may gather the values locally, load the values from a storage device or request the values from an external entity. For parameters such as the Natural Environment, the intermediary may be able to deduce the information on its own, especially when the intermediary is located near the content user. For parameters such as the User Preferences, the human user may have registered a set of profiles to be stored at the intermediary.

[0067] The invention allows intermediate network elements along a content flow path to actively collaborate their content delivery efforts to enhance user experience in content retrieval. With more and more content delivery networks (CDN) deployed in the Internet, the invention disclosed in this document allows intermediaries of such CDNs to orchestrate their efforts by the provision of distributed rules. Rules loaded on one network element can be distributed to other intermediaries in real time, so that adaptation of data contents and contents request can be performed at a more suitable node. It also allows better control over the operations of content delivery.

[0068] In addition, the disclosed invention contains methods and means which are specific to the real-time delivery of Audio Visual content in a packet switch network. This allows rule authors to create rules that can react to fluctuations in the network conditions of the content streaming more speedily. Authors can also create rules that are based on the device capabilities and user preferences of the content consumers. When such rules are authored carefully with suitable adaptation services, the overall user experience in content retrieval will be significantly enhanced.

Claims

1. A network control framework apparatus for controlling resources at an intermediate network element connecting two or more communications networks comprising:

a) a gateway module (101) providing gateway functionality,

b) a rule engine module (102) to perform network resource control decision based on specified rules, wherein the rules are specified in a rule specification format hereafter referred to as a Rule Specification,

c) at least one special package (103) added on to the rule engine module offering specialized functionality to the rule engine module,

d) a rule injection module (104) to Inject or remove Rule Specification to or from the rule engine module, and

e) a means for distribution of said Rule Specification to at least one intermediate network element comprising

i. means for distribution of indications in the Rule Specification to indicate that part or whole of the Rule Specification is to be distributed,

ii. means for distribution of a signature embedded into data packets to announce the capabilities of the intermediate network elements the data packet traversed,

iii. means for parsing the Rule Specification to determine if part or whole of the specified Rule Specification is distributed,

iv. means for identifying the target network element to distribute part or whole of a Rule Specification,

v. means for distribution of a signalling embedded into data packets to inform target network element of the distribution of part or whole of Rule Specification,

vi. means for retrieval of the part or whole of Rule Specification distributed to the target network element from the intermediate network element that distributes the part or whole of Rule Specification.

2. The apparatus as recited in claim 1, wherein the format of said indications of part or whole of Rule Specification for distribution comprises

i. the specification of the direction of distribution by specifying the endpoint of the specified direction,

ii. the specification of the number of intermediate network elements towards the specified endpoint,

iii. the specification of the number of intermediate network elements from the specified endpoint, and/or

iv. the specific content distributed at the intermediate network elements.

3. The apparatus as recited in claim 1, wherein the format of said signature embedded into data packets comprises

i. the identification of the intermediate network element the signature belongs to;

ii. the special packages that are installed on the intermediate network element the signature belongs to, and

iii. the capability of accepting or generating part or whole of a Rule Specifications for distribution.

4. The apparatus as recited in claim 1 or 3 wherein the signatures of the intermediate network elements that the data packets traversed are stored with the starting and ending points between which the data packets traversed in the order of which the data packets traversed and the transmission protocol the data packets belongs to.

5. The apparatus as recited in claim 1, 3 or 4, wherein the format of said signature comprises the identification of the intermediate network element and the installed at least one special package at the intermediate network element.

6. The apparatus as recited in claims 1, 3, 4 or 5, wherein the format of said signatures comprises

i. the identification of the ending point that the data packets flow to,

ii. the identification of the starting point that the data packets flow from,

iii. the transmission protocol the data packets belongs to,

iv. the array of signatures of the intermediate network elements in the order of the data packets traverse from the intermediate network element where the data format is stored to the ending point, and

v. the number of signatures of the intermediate network elements in the order of the data packets traverse from the intermediate network element where the data format is stored to the ending point.

7. The apparatus as recited in any of the preceding claims, further comprising means for signalling to signal the intermediate network element to express the desire to distribute collection of rules in a Rule Specification to the intermediate network element comprising ,

i. the identification of the intermediate network element where the collection of rules in a Rule Specification is distributed to,

ii. the identification of the intermediate network element where the collection of the at least one rules In a Rule Specification is distributed from, and

iii. the identification of the collection of the at least one rule in a Rule Specification.

8. The apparatus as recited In any of the preceding claims, further comprising a means of retrieving the collection of rules in a Rule Specification from the intermediate network element that distributes the collection of rules by the intermediate network element where the collection of rules is distributed to, comprising

i. means for establishing a communication channel between the intermediate network element where the collection of rules is distributed to and the intermediate network element where the collection of rules is distributed from,

ii. means for providing the identification of the collection of rules that is distributed via the communications channel by the intermediate network element where the collection of rules is distributed to, and

iii. means for transmitting the collection of rules that is distributed via the communications channel by the intermediate network element where the collection of rules is distributed from.

9. The apparatus as recited in any of the preceding claims, wherein said communications networks comprise an endpoint node, hereafter referred to as a client node, for sending a request to the other endpoint node, hereafter referred to as a server node, via at least one intermediate network element, wherein the server node is adapted for accepting the request with an appropriate response, wherein said communications networks further comprise means for setting up a communications channel between the server node and the client node through the intermediate network elements, and wherein the server node is adapted for starting transmitting data packets through the communications channel to the client node until the client node sends a request, via the intermediate network elements, to tear down the communications channel, and wherein the client node is adapted for transmitting information about the transmission statistics back to the server node.

10. The apparatus as recited in claim 9, further comprising a means of providing the author of Rule Specification to trigger a singular or plurality of rules at a intermediate network element based on the following control methods

i. the rule to be evaluated when the intermediate network element received a request packet from the client node to the server node,

ii. the rule to be evaluated when the intermediate network element received a response packet from the server node to the client node,

iii. the rule to be evaluated when the intermediate network element received a data packet containing contents sent by the server node to the client node through the communications channel established between the server node and the client node,

iv. the rule to be evaluated when the intermediate network element received a data packet containing the transmission statistics from the client node to the server node,

v. the rule to be evaluated when the intermediate network element received a specified number of data packet containing contents sent by the server node to the client node through the communications channel established between the server node and the client node, and

vi. the rule to be evaluated when the intermediate network element received a data packet containing contents sent by the server node to the client node through the communications channel established between the server node and the client node after the elapse of a recurrent timer of a specified timer value.

11. The apparatus as recited in any of the preceding claims comprising a controi means for using a set of parameters in the Rule Specification to control at least one content or content delivery sessions to achieve device independence in the delivery of said content, comprising

i. the set of User Preference parameters consisting of the preferences of the human user consuming the content,

ii. the set of Agent Capabilities parameters consisting of the capabilities of the software agent employed by the human user to retrieve the content,

iii. the set of Device Capabilities parameters consisting of the capabilities of the hardware employed by the human user to retrieve the content, and

iv. the set Natural Environment parameters consisting of the information about the environment in which the human user retrieves the content

12. The apparatus as recited in claim 13, wherein the set of User Preference parameters comprises

i. the human user's preferences on the method of retrieving the content,

ii. the human user's preferences on the language used in the retrieved contehts,

iii. the human user's preferences on the presentation of the retrieved content,

iv. the age group of the human user retrieving the content,

v. the gender of the human user retrieving the content, and

vi. the employment status of the human user retrieving the content.

13. The apparatus as recited in claim 11, wherein the set of Agent Capabilities parameters comprises

L the type of software agent employed by the human user to retrieve the content,

ii. the content formats supported by the software agent employed by the human user to retrieve the content,

iii. the content languages supported by the software agent employed by the human user to retrieve the content, and

iv. the transmission protocols supported by the software agent employed by the human user to retrieve the content.

14. The apparatus as recited in claim 11, wherein the set of Device capabilities parameters comprises

i. the type of hardware employed by the human user to retrieve the content,

ii. the processor speed and processor family of the hardware employed by the human user to retrieve the content,

iii. the memory capacity of the physical and secondary storage of the hardware employed by the human user to retrieve the content,

iv. the display depth and resolution of the hardware employed by the human user to retrieve the content, and

v. the operating system running on the hardware employed by the human user to retrieve the content.

15. The apparatus as recited in claim 11, wherein the set of Natural Environment parameters comprising

i. the information of the location where the human user is retrieving the content,

ii. the information of the mobility of the human user retrieving the content, and

iii. the information of the illuminations conditions in which the human user is retrieving the content.

16. The apparatus as recited in any of claims 11 to 14, wherein the at least one special package is capable of interpreting and evaluating said Rule Specification.

17. A network control framework method for controlling resources at an intermediate network element connecting two or more communications networks comprising the steps of:

a) providing gateway functionality by a gateway module,

b) performing network resource control decision by a rule engine module based on specified rules, wherein the rules are specified in a rule specification format hereafter referred to as a Rule Specification,

c) offering specialized functionality to the rule engine module by at least one special package added on to the rule engine module,

d) injecting or removing Rule Specification to or from the rule engine module by a rule injection module, and

e) distribution of said Rule Specification to at least one intermediate network element comprising the steps of

i. distribution of indications in the Rule Specification to indicate that part or whole of the Rule Specification is to be distributed,

ii. distribution of a signature embedded into data packets to announce the capabilities of the intermediate network elements the data packet traversed,

iii. parsing the Rule Specification to determine if part or whole of the specified Rule Specification is distributed,

iv. identifying the target network element to distribute part or whole of a Rule Specification,

v. distribution of a signalling embedded into data packets to inform target network element of the distribution of part or whole of Rule Specification,

vi. retrieval of the part or whole of Rule Specification distributed to the target network element from the intermediate network element that distributes the part or whole of Rule Specification.

18. The method as recited in claim 17, further comprising a step of extracting the signature of intermediate network elements embedded in at least one data packet, comprising the steps of

i. checking if there are embedded signatures in the data packets,

ii. checking if there exist a signature in a predetermined data format that is previously stored having the same starting and ending points and transmission protocol,

iii. allocating a new data format when there is no data format that is previously stored having the same starting and ending points and transmission protocol,

iv. purging data stored in the data format that previously existed having the same starting point, ending point and transmission protocol,

v. preparing an empty last-in-first-out data structure,

vi. extracting each embedded signature in the data packet and pushing it to the last-in-first-out data structure,

vii. removing each element in the last-in-first-out data structure and recording it to the predetermined data format, and

viii. recording the number of embedded signature extracted in the predetermined data format.

19. The method as recited in one of claims 17 or 18, further comprising a step of parsing a Rule Specification to determine if part or whole of the Rule Specification is to be distributed comprising the steps of

i. checking each rule in the Rule Specification for syntactical validity,

ii. rejecting the rule if there is syntactical errors,

iii. checking the rule for a distribution indication,

iv. evaluating the rule locally if there exist no distribution indication,

v. determining the remote intermediate network element to distribute the rule to,

vi. evaluating the rule locally if no suitable remote intermediate network element to distribute the rule to can be found,

vii. checking if the remote intermediate network element contains the special package or special packages required in the rule,

viii. evaluating the rule locally if the remote intermediate network element do not have the required special package or special packages, and

ix. distributing the rule to the remote intermediate network element.

20. The method as recited in claim 17, further comprising a method of determining the remote intermediate network element that a rule is to be distributed to given a predetermined distribution indication, comprising the steps of

i. locating a signature in a predetermined data format with the matching starting point, ending point and transmission protocol,

ii. declaring no suitable remote intermediate network element if no predetermined data format can be located,

iii. setting a temporary variable to the specified number of the intermediaries towards or from the specified endpoint in the given distribution indication,

iv. setting the temporary variable to the value of the number of intermediaries as given in the located predetermined data format if the specified number of the intermediate network elements towards or from the specified endpoint In the given distribution indication is greater than the number of intermediate network elements towards or from the specified ending point in the given distribution indication,

v. whereas the specified distribution indication consists of the specification of the ending point and the specification of the number of intermediate network elements towards the specified ending point, set the temporary variable to a value equals the number of intermediate network elements given in the located predetermined data format minus the original value in the temporary variable,

vi. whereas the specified distribution indication consists of the specification of the ending point and the specification of the number of intermediate network elements from the specified ending point, set the temporary variable to a value equals the original value in the temporary variable minus 1,

vii. declaring the remote intermediate network element to be the network element specified in a signature stored in the located predetermined data format where the signature has an index in the array of signatures in the located predetermined data format equals to the value stored in the temporary variable should such an index exist, and

viii. declaring no suitable remote intermediate network element should the index equal to the value stored in the temporary variable does not exist in the array of signatures in the located predetermined data format.

21. A communications network comprising at network control framework apparatus as recited in any of claims 1 to 16 for controlling resources at an intermediate network element connecting two or more communications networks.

Patentansprüche

1. Netzwerksteuergrundstrukturvorrichtung zum Steuern von Ressourcen an einem zwischengeschalteten Netzwerkelement, welches zwei oder mehr Kommunikationsnetzwerke verbindet, mit:

a) einem Gatewaymodul (101), welches Gatewayfunktionalität zur Verfügung stellt,

b) einem Regelverarbeitungsmodul (102), um auf der Basis von spezifizierten Regeln eine Netzwerkressourcensteuerentscheidung zu erbringen, wobei die Regeln in einem Regelspezifikationsformat spezifiziert sind, welches im folgenden als eine Regelspezifikation bezeichnet ist,

c) mindestens einem speziellen Paket (103), welches dem Regelverarbeitungsmodul hinzugefügt ist, dem Regelverarbeitungsmodul spezialisierte Funktionalität anbietend,

d) einem Regelinjektionsmodul (104), um eine Regelspezifikation in das Regelbearbeitungsmodul zu injizieren oder aus dem Regelbearbeitungsmodul zu entfernen, und

e) einem Mittel zum Verteilen der Regelspezifikation an mindestens ein zwischengeschaltetes Netzwerkelement, aufweisend

i. Mitteln zum Verteilen von Hinweisen in der Regelspezifikation, um anzuzeigen, dass ein Teil der Regelspezifikation oder die ganze Regelspezifikation zu verteilen ist,

ii. Mitteln zum Verteilen einer in Datenpaketen eingebetteten Signatur, um die Fähigkeiten der zwischengeschalteten Netzwerkelemente zu melden, die die Datenpakete durchquert haben,

iii. Mitteln zum Durchführen eines Parsing der Regelspezifikation, um zu bestimmen, ob ein Teil der spezifizierten Regelspezifikation oder die ganze spezifizierte Regelspezifikation verteilt wird,

iv. Mitteln zum Identifizieren des Zielnetzwerkelements, um einen Teil einer Regelspezifikation oder eine ganzen Regelspezifikation zu verteilen,

v. Mitteln zum Verteilen einer in Datenpakete eingebetteten Signalgebung, um ein Zielnetzwerkelement über die Verteilung eines Teils einer Regelspezifikation oder einer ganzen Regelspezifikation zu informieren,

vi. Mitteln zum Wiedergewinnen des Teils der Regelspezifikation oder der ganzen Regelspezifikation, die an das Zielnetzwerkelement verteilt wurde, von dem zwischengeschalteten Netzwerkelement, welches den Teil der Regelspezifikation oder die ganze Regelspezifikation verteilt hat.

2. Vorrichtung nach Anspruch 1, wobei das Format der Hinweise zur Verteilung eines Teils einer Regelspezifikation oder einer ganzen Regelspezifikation aufweist

i. Mittel zum Verteilen Hinweisen in der Regelspezifikation, um anzuzeigen, dass ein Teil der Regelspezifikation oder die ganze Regelspezifikation zu verteilen ist,

ii. die Spezifikation der Richtung einer Verteilung durch Spezifizieren des Endpunkts der spezifizierten Richtung,

iii. die Spezifikation der Anzahl der zwischengeschalteten Netzwerkelemente in Richtung auf den spezifizierten Endpunkt,

iv. die Spezifikation der Anzahl der zwischengeschalteten Netzwerkelemente von dem spezifizierten Endpunkt, und/oder

v. der spezifische Inhalt, der an den zwischengeschalteten Netzwerkelementen verteilt wird.

3. Vorrichtung nach Anspruch 1, wobei das Format der in den Datenpaketen eingebetteten Signatur aufweist

i. die Identifikation des zwischengeschalteten Netzwerkelements, zu der die Signatur gehört,

ii. die speziellen Pakete, die auf dem zwischengeschalteten Netzwerkelement, zu dem die Signatur gehört, installiert sind, und

iii. die Fähigkeit eines Akzeptierens oder eines Erzeugens von einem Teil einer Regelspezifikation oder einer ganzen Regelspezifikation für eine Verteilung.

4. Vorrichtung nach den Ansprüchen 1 oder 3, wobei die Signaturen der zwischengeschalteten Netzwerkelemente, die die Datenpakete durchquert haben, mit den Start- und Endpunkten gespeichert werden, zwischen denen die Datenpakete gewandert sind, in der Reihenfolge, in der die Datenpakete gewandert sind, und des Übertragungsprotokolls, zu dem die Datenpakete gehören.

5. Vorrichtung nach den Ansprüchen 1, 3 oder 4, wobei das Format der Signatur die Identifikation des zwischengeschalteten Netzwerkelements und des installierten mindestens einen speziellen Pakets an dem zwischengeschalteten Netzwerkelement aufweist.

6. Vorrichtung nach den Ansprüchen 1, 3, 4 oder 5, wobei das Format der Signaturen aufweist

i. die Identifikation des Endpunkts, zu welchem die Datenpakete fließen,

ii. die Identifikation des Startpunkts, von dem die Datenpakete fließen,

iii. das Übertragungsprotokoll, zu dem die Datenpakete gehören,

iv. die Anordnung von Signaturen der zwischengeschalteten Netzwerkelemente in der Ordnung, in der die Datenpakete von dem zwischengeschalteten Netzwerkelement, wo das Datenformat gespeichert ist, zu dem Endpunkt wandern, und

v. die Anzahl von Signaturen der zwischengeschalteten Netzwerkelemente in der Ordnung, in der die Datenpakete von dem zwischengeschalteten Netzwerkelement, wo das Datenformat gespeichert ist, zu dem Endpunkt wandern.

7. Vorrichtung nach einem der vorstehenden Ansprüche, weiterhin aufweisend Mittel zur Signalgebung, um dem zwischengeschalteten Netzwerkelement ein Signal zu geben, um den Wunsch auszudrücken, eine Sammlung von Regeln in einer Regelspezifikation an das zwischengeschaltete Netzwerkelement zu verteilen, aufweisend

i. die Identifikation des zwischengeschalteten Netzwerkelements, wohin die Sammlung von Regeln in einer Regelspezifikation verteilt wird,

ii. die Identifikation des zwischengeschalteten Netzwerkelements, von wo aus die Sammlung per mindestens einer Regel in einer Regelspezifikation verteilt wird, und

iii. die Identifikation der Sammlung der mindestens einen Regel in einer Regelspezifikation.

8. Vorrichtung nach einem der vorstehenden Ansprüche, weiterhin aufweisend ein Mittel zum Wiedergewinnen der Sammlung von Regeln in einer Regelspezifikation von dem zwischengeschalteten Netzwerkelement, welches die Sammlung von Regeln verteilt, durch das zwischengeschaltete Netzwerkelement, wohin die Sammlung von Regeln verteilt ist, aufweisend

i. Mittel zum Etablieren eines Kommunikationskanals zwischen dem zwischengeschalteten Netzwerkelement, wohin die Sammlung von Regeln verteilt ist, und dem zwischengeschalteten Netzwerkelement, von wo die Sammlung von Regeln verteilt wird,

ii. Mittel zum Liefern der Identifikation der Sammlung von Regeln, die über die Kommunikationskanäle verteilt ist, durch das zwischengeschaltete Netzwerkelement, wohin die Sammlung von Regeln verteilt ist, und

iii. Mittel zum Übertragen der Sammlung von Regeln, welche über die Kommunikationskanäle verteilt ist, durch das zwischengeschaltete Netzwerkelement, von wo die Sammlung von Regeln verteilt wird.

9. Vorrichtung nach einem der vorstehenden Ansprüche, wobei die Kommunikationsnetzwerke einen im folgenden als einen Clientknoten bezeichneten Endpunktknoten aufweisen, um eine Anfrage zu den im folgenden als einen Serverknoten bezeichneten anderen Endpunktknoten über mindestens ein zugeschaltetes Netzwerkelement zu senden, wobei der Serverknoten angepasst ist, um die Anfrage mit einer angemessenen Antwort zu akzeptieren, wobei die Kommunikationsnetzwerke weiterhin Mittel zum Einstellen eines Kommunikationskanals zwischen dem Serverknoten und dem Clientknoten durch die zwischengeschalteten Netzwerkelemente aufweisen, und wobei der Serverknoten angepasst ist, um eine Übertragung von Datenpaketen durch den Kommunikationskanal zu dem Clientknoten zu beginnen, bis der Clientknoten über die zwischengeschalteten Netzwerkelemente eine Anfrage sendet, um den Kommunikationskanal zu schließen, und wobei der Clientknoten angepasst ist, um Information über die Übertragungsstatistiken zurück zu dem Serverknoten zu übertragen.

10. Vorrichtung nach Anspruch 9, weiterhin aufweisend ein Mittel zum Liefern des Autors einer Regelspezifikation, um eine einzelne oder eine Mehrzahl von Regeln an einem zwischengeschalteten Netzwerkelement zu triggern, auf der Basis der folgenden Steuerverfahren

i. die Regel, die auszuwerten ist, wenn das zwischengeschaltete Netzwerkelement ein Anfragepaket von dem Clientknoten zu dem Serverknoten empfangen hat,

ii. die Regel, die auszuwerten ist, wenn das zwischengeschaltete Netzwerkelement ein Antwortpaket von dem Serverknoten zu dem Clientknoten empfangen hat,

iii. die Regel, die auszuwerten ist, wenn das zwischengeschaltete Netzwerkelement ein Datenpaket empfangen hat, welches Inhalte enthält, welche durch den Serverknoten zu dem Clientknoten durch den zwischen dem Serverknoten und dem Clientknoten etablierten Kommunikationskanal gesandt wurden,

iv. die Regel, die auszuwerten ist, wenn das zwischengeschaltete Netzwerkelement ein die Übertragungsstatistiken von dem Clientknoten zu dem Serverknoten enthaltendes Datenpaket empfangen hat,

v. die Regel, die auszuwerten ist, wenn das zwischengeschaltete Netzwerkelement eine spezifizierte Anzahl von Datenpaketen erhalten hat, die Inhalte enthalten, die durch den Serverknoten zu dem Clientknoten durch den zwischen dem Serverknoten und dem Clientknoten etablierten Kommunikationskanal gesandt wurden, und

vi. die Regel, die auszuwerten ist, wenn das zwischengeschaltete Netzwerkelement ein Datenpaket empfangen hat, welches Inhalte enthält, die durch den Serverknoten durch den zwischen dem Serverknoten und dem

Clientknoten etablierten Kommunikationskanal zu dem Clientknoten gesandt wurden, nach dem Ablauf eines periodischen Zeitgebers eines spezifizierten Zeitgeberwertes.

11. Vorrichtung nach einem der vorstehenden Ansprüche, mit einem Steuermittel zur Verwendung eines Satzes von Parametern in der Regelspezifikation, um mindestens einen Inhalt oder Inhaltsauslieferungssitzungen zu steuern, um eine Bauelementunabhängigkeit in der Auslieferung des Inhalts zu erreichen, aufweisend

i. den Satz von Parametern von Benutzervorlieben, bestehend aus den Vorlieben des menschlichen Benutzers, der den Inhalt konsumiert,

ii. den Satz von Parametern von Agentenfähigkeiten, der aus den Fähigkeiten des Softwareagenten besteht, der durch den menschlichen Benutzer verwendet wird, um den Inhalt zu gewinnen,

iii. den Satz von Parametern von Bauelementfähigkeiten, der aus den Fähigkeiten der Hardware besteht, die durch den menschlichen Benutzer verwendet wird, um den Inhalt zu gewinnen, und

iv. den Satz von Parametern der natürlichen Umgebung, der aus der Information über die Umgebung besteht, in welcher der menschliche Benutzer den Inhalt gewinnt.

12. Vorrichtung nach Anspruch 13, wobei der Satz von Parametern von Benutzervorlieben aufweist

i. die Vorlieben des Benutzers bezüglich des Verfahrens des Gewinnens des Inhalts,

ii. die Vorlieben des Benutzers bezüglich der in den gewonnenen Inhalten verwendeten Sprache,

iii. die Vorlieben des Benutzers bezüglich der Präsentation des gewonnenen Inhalts,

iv. die Altersgruppe des den Inhalt gewinnenden menschlichen Benutzers,

v. das Geschlecht des den Inhalt gewinnenden menschlichen Benutzers, und

vi. der Beschäftigungsstatus des den Inhalt gewinnenden menschlichen Benutzers.

13. Vorrichtung nach Anspruch 11, wobei der Satz von Parametern von Agentenfähigkeiten aufweist

i. den Typ des Softwareagenten, der durch den menschlichen Benutzer verwendet wird, um den Inhalt zu gewinnen,

ii. die Inhaltsformate, die durch den Softwareagenten unterstützt werden, der durch den menschlichen Benutzer verwendet wird, um den Inhalt zu gewinnen,

iii. die Inhaltssprachen, die durch den Softwareagenten unterstützt werden, der durch den menschlichen Benutzer verwendet wird, um den Inhalt zu gewinnen, und

iv. die Übertragungsprotokolle, die durch den Softwareagenten unterstützt werden, der durch den menschlichen Benutzer verwendet werden, um den Inhalt zu gewinnen.

14. Vorrichtung nach Anspruch 11, wobei der Satz von Parametern von Bauelementfähigkeiten aufweist

i. den Typ der Hardware, die durch den menschlichen Benutzer verwendet wird, um den Inhalt zu gewinnen,

ii. die Prozessorgeschwindigkeit und Prozessorfamilie der Hardware, die von dem menschlichen Benutzer verwendet wird, um den Inhalt zu gewinnen,

iii. die Speicherkapazität des physikalischen und sekundären Speichers der Hardware, die von dem menschlichen Benutzer verwendet wird, um den Inhalt zu gewinnen,

iv. die Anzeigetiefe und -auflösung der Hardware, die von dem menschlichen Benutzer verwendet wird, um den Inhalt zu gewinnen, und

v. das Betriebssystem, welches auf der Hardware läuft, die von dem menschlichen Benutzer verwendet wird, um den Inhalt zu gewinnen.

15. Vorrichtung nach Anspruch 11, wobei der Satz von Parametern über die natürliche Umgebung aufweist

i. die Information bezüglich des Ortes, wo der menschliche Benutzer den Inhalt gewinnt,

ii. die Information bezüglich der Mobilität des menschlichen Benutzers, der den Inhalt gewinnt, und

iii. die Information bezüglich der Beleuchtungsbedingungen, unter welchen der menschliche Benutzer den Inhalt gewinnt.

16. Vorrichtung nach einem der Ansprüche 11 bis 14, wobei das mindestens eine spezielle Paket in der Lage ist, die Regelspezifikation zu interpretieren und zu ermitteln.

17. Netzwerksteuergrundstrukturverfahren zum Steuern von Ressourcen an einem zwischengeschalteten Netzwerkelement, welches zwei oder mehr Kommunikationsnetzwerke verbindet, mit den Schritten:

a) Liefern einer Gatewayfunktionalität durch ein Gatewaymodul,

b) Liefern einer Netzwerkressourcensteuerentscheidung durch ein Regelverarbeitungsmodul auf der Basis von spezifizierten Regeln, wobei die Regeln in einem Regelspezifikationsformat spezifiziert sind, welches im folgenden als eine Regelspezifikation bezeichnet wird,

c) dem Regelverarbeitungsmodul wird durch mindestens ein spezielles Paket, welches dem Regelverarbeitungsmodul hinzugefügt wird, eine spezialisierte Funktionalität angeboten,

d) eine Regelspezifikation wird den Regelverarbeitungsmodul durch ein Regelinjektionsmodul injiziert oder von diesem entfernt, und

e) Verteilung der Regelspezifikation zu mindestens einem zwischengeschalteten Netzwerkelement, mit den Schritten

i. Verteilung von Hinweisen in der Regelspezifikation, um anzuzeigen, dass ein Teil der Regelspezifikation oder die ganze Regelspezifikation zu verteilen ist,

ii. Verteilung einer Signatur, eingebettet in Datenpaketen, um auf die Fähigkeiten der zwischengeschalteten Netzwerkelemente hinzuweisen, die das Datenpaket durchquert hat,

iii. Durchführen eines Parsing der Regelspezifikation, um zu bestimmen, ob ein Teil der spezifizierten Regelspezifikation oder die ganze spezifizierte Regelspezifikation verteilt ist,

iv. Identifizieren des Zielnetzwerkelements, um einen Teil einer Regelspezifikation oder die ganze Regelspezifikation zu verteilen,

v. Verteilung einer Signalgebung, eingebettet in Datenpakete, um ein Zielnetzwerkelement über die Verteilung eines Teils einer Regelspezifikation oder einer ganzen Regelspezifikation zu informieren,

vi. Gewinnen des Teils der Regelspezifikation oder der ganzen Regelspezifikation, die zu dem Zielnetzwerkelement verteilt wurde, von dem zwischengeschalteten Netzwerkelement, welches den Teil der Regelspezifikation oder die ganze Regelspezifikation verteilt.

18. Verfahren nach Anspruch 17, weiterhin einen Schritt des Extrahierens der Signatur von zwischengeschalteten Netzwerkelementen aufweisend, die in mindestens einem Datenpaket eingebettet ist, mit den Schritten

i. Prüfen, ob sich in den Datenpaketen eingebettete Signaturen befinden,

ii. Prüfen, ob eine Signatur in einem vorbestimmten Datenformat existiert, das zuvor mit den gleichen Start- und Endpunkten und Übertragungsprotokoll gespeichert wurde,

iii. Zuordnen eines neuen Datenformats, wenn kein Datenformat existiert, welches zuvor mit den gleichen Start- und Endpunkten und Transmissionsprotokoll gespeichert wurde,

iv. Löschen von den in dem Datenformat gespeicherten Daten, die zuvor mit dem gleichen Startpunkt, Endpunkt und Transmissionsprotokoll existierten,

v. Präparieren einer leeren last-in-first-out Datenstruktur,

vi. Extrahieren jeder eingebetteten Signatur in dem Datenpaket und Einschieben derselben in die last-in-first-out Datenstruktur,

vii. Entfernen jedes Elements in der last-in-first-out Datenstruktur und Aufzeichnen derselben in dem vorbestimmten Datenformat, und

viii. Aufzeichnen der Anzahl eingebetteter Signaturen, die in dem vorbestimmten Datenformat extrahiert wurden.

19. Verfahren nach einem der Ansprüche 17 oder 18, weiterhin einen Schritt einer Durchführung eines Parsing einer Regelspezifikation aufweisend, um zu bestimmen, ob ein Teil der Regelspezifikation oder die ganze Regelspezifikation zu verteilen ist, mit den Schritten

i. Prüfen jeder Regel in der Regelspezifikation auf ihre Gültigkeit hinsichtlich einer Syntax,

ii. Ausstoßen der Regel, wenn Syntaxfehler vorhanden sind,

iii. Prüfen der Regel bezüglich eines Hinweises auf Verteilung,

iv. lokales Ermitteln der Regel, wenn kein Hinweise auf eine Verteilung existiert,

v. Bestimmen des entfernten zwischengeschalteten Netzwerkelements, um die Regel dorthin zu verteilen,

vi. lokales Ermitteln der Regel, wenn kein passendes entferntes zwischengeschaltetes Netzwerkelement gefunden werden kann, um die Regel zu verteilen,

vii. Prüfen, ob das entfernte zwischengeschaltete Netzwerkelement das spezielle Paket oder spezielle Pakete aufweist, die in der Regel benötigt werden,

viii. lokales Ermitteln der Regel, wenn das entfernte zwischengeschaltete Netzwerkelement das benötigte spezielle Paket oder die benötigten speziellen Pakete nicht aufweist, und

ix. Verteilen der Regel zu dem entfernten zwischengeschalteten Netzwerkelement.

20. Verfahren nach Anspruch 17, weiterhin aufweisend ein Verfahren, um das entfernte zwischengeschaltete Netzwerkelement zu bestimmen, dass eine Regel zu diesem zu verteilen ist, einen vorbestimmten Verteilungshinweis vorausgesetzt, mit den Schritten

i. Lokalisieren einer Signatur in einem vorbestimmten Datenformat mit dem passenden Startpunkt, Endpunkt und Übertragungsprotokoll,

ii. es wird keine passendes entferntes zwischengeschaltetes Netzwerkelement erklärt, wenn kein vorbestimmtes Datenformat lokalisiert werden kann,

iii. Einstellen einer temporären Variable auf die spezifizierte Anzahl der Zwischenstationen in Richtung auf oder von dem spezifizierten Endpunkt in dem vorgegebenen Verteilungshinweis,

iv. Einstellen der temporären Variable auf den Wert der Anzahl von Zwischenstationen, wie in dem lokalisierten vorbestimmten Datenformat vorgegeben, wenn die spezifizierte Anzahl von zwischengeschalteten Netzwerkelementen auf den spezifizierten Endpunkt in dem vorgegebenen Verteilungshinweis oder von dem spezifizierten Endpunkt aus in dem vorgegebenen Verteilungshinweis größer ist als die Anzahl von zwischengeschalteten Netzwerkelementen in Richtung auf den spezifizierten Endpunkt in dem vorgegebenen Verteilungshinweis oder von dem spezifizierten Endpunkt aus in dem vorgegebenen Verteilungshinweis,

v. wobei der spezifizierte Verteilungshinweis aus der Spezifikation des Endpunkts und der Spezifikation der Anzahl von zwischengeschalteten Netzwerkelementen in Richtung auf den spezifizierten Endpunkt besteht, wobei die temporäre Variable auf einen Wert gesetzt wird, der der Anzahl der zwischengeschalteten Netzwerkelemente entspricht, die in dem lokalisierten vorbestimmten Datenformat vorgegeben sind, minus dem Originalwert in der temporären Variablen,

vi. wobei der spezifizierte Verteilungshinweis aus der Spezifikation des Endpunkts und der Spezifikation der Anzahl von zwischengeschalteten Netzwerkelementen von dem spezifizierten Endpunkt aus besteht, wobei die temporäre Variable auf einen Wert gesetzt wird, der dem Originalwert in der temporären Variable minus 1 entspricht,

vii. Erklären des entfernten zwischengeschalteten Netzwerkelementes als das Netzwerkelement, welches in einer Signatur spezifiziert ist, die in dem lokalisierten vorbestimmten Datenformat gespeichert ist, wobei die Signatur einen Index in der Anordnung von Signaturen in dem lokalisierten vorbestimmten Datenformat aufweist, welcher dem Wert entspricht, der in der temporären Variable gespeichert ist, falls ein solcher Index existiert, und

viii. es wird kein passendes entferntes zwischengeschaltetes Netzwerkelement erklärt, sollte der Index, der gleich dem Wert ist, der in der temporären Variable gespeichert ist, nicht in der Anordnung von Signaturen in dem lokalisierten vorbestimmten Datenformat existieren.

21. Kommunikationsnetzwerk mit einer Netzwerksteuergrundstrukturvorrichtung nach einem der Ansprüche 1 bis 16 zum Steuern von Ressourcen an einem zwischengeschalteten Netzwerkelement, welches zwei oder mehr Kommunikationsnetzwerke verbindet.

Revendications

1. Dispositif de structure de commande de réseau pour commander des ressources au niveau d'un élément de réseau intermédiaire connectant deux réseaux de communications ou plus comprenant :

a) un module (101) de passerelle fournissant une fonctionnalité de passerelle,

b) un module (102) de moteur de règles pour réaliser une décision de commande de ressources de réseau sur la base de règles spécifiées, dans lequel les règles sont spécifiées dans un format de spécification de règles ci-après dénommé Spécification de Règles,

c) au moins un progiciel (103) spécial ajouté sur le module de moteur de règles offrant une fonctionnalité spécialisée au module de moteur de règles,

d) un module (104) d'injection de règles pour injecter dans ou retirer une Spécification de Règles du module de moteur de règles, et

e) un moyen pour répartir ladite Spécification de Règles vers au moins un élément de réseau intermédiaire comprenant

i. un moyen pour répartir des indications dans la Spécification de Règles pour indiquer qu'une partie ou la totalité de la Spécification de Règles va être répartie,

ii. un moyen pour répartir une signature intégrée dans des paquets de données pour annoncer les possibilités des éléments de réseau intermédiaire que le paquet de données a traversé,

iii. un moyen pour faire l'analyse de la Spécification de Règles pour déterminer si une partie ou la totalité de la Spécification de Règles spécifiées est répartie,

iv. un moyen pour identifier l'élément de réseau cible pour répartir une partie ou la totalité d'une Spécification de Règles,

v. un moyen pour répartir une signalisation intégrée dans des paquets de données pour informer un élément de réseaux cibles de la répartition d'une partie ou de la totalité d'une Spécification de Règles,

vi. un moyen pour une récupération de la partie ou de la totalité d'une Spécification de Règles répartie vers l'élément de réseau cible depuis l'élément de réseau intermédiaire qui répartit la partie ou la totalité de la Spécification de Règles.

2. Dispositif selon la revendication 1, dans lequel le format desdites indications de la partie ou de la totalité d'une Spécification de Règles pour une répartition comprend :

i. la spécification de la direction de répartition par la spécification du point d'extrémité de la direction spécifiée,

ii. la spécification du nombre d'éléments de réseau intermédiaire vers le point d'extrémité spécifié,

iii. la spécification du nombre d'éléments de réseau intermédiaire depuis le point d'extrémité spécifié, et/ou

iv. le contenu spécifique réparti au niveau des éléments de réseau intermédiaire.

3. Dispositif selon la revendication 1, dans lequel le format de ladite signature intégrée dans des paquets de données comprend

i. l'identification de l'élément de réseau intermédiaire auquel appartient la signature,

ii. les progiciels spéciaux qui sont installés sur l'élément de réseau intermédiaire auquel appartient la signature et

iii. la possibilité d'accepter ou de générer une partie ou la totalité de Spécification de Règles pour une répartition.

4. Dispositif selon la revendication 1 ou 3 dans lequel les signatures des éléments de réseau intermédiaire que les paquets de données ont traversés sont mémorisées avec les points de départ et d'arrivée que les paquets de données ont traversés dans l'ordre selon lequel les paquets de données ont traversé et le protocole de transmission auquel appartiennent les paquets de données.

5. Dispositif selon la revendication 1, 3 ou 4, dans lequel le format de ladite signature comprend l'identification de l'élément de réseau intermédiaire et le au moins un progiciel spécial installé au niveau de l'élément de réseau intermédiaire.

6. Dispositif selon la revendication 1, 3, 4 ou 5, dans lequel le format desdites signatures comprend

i. l'identification du point d'arrivée vers lequel les paquets de données circulent,

ii. l'identification du point de départ depuis lequel les paquets de données circulent,

iii. le protocole de transmission auquel les paquets de données appartiennent,

iv. l'ensemble des signatures des éléments de réseau intermédiaire dans l'ordre de traversée des paquets de données depuis l'élément de réseau intermédiaire où le format de données est mémorisé jusqu'au point d'arrivée, et

v. le nombre de signatures des éléments de réseau intermédiaire dans l'ordre de la traversée des paquets de données depuis l'élément de réseau intermédiaire où le format de données est mémorisé jusqu'au point d'arrivée.

7. Dispositif selon l'une quelconque des revendications précédentes, comprenant en outre un moyen pour une signalisation pour signaler à l'élément de réseau intermédiaire d'exprimer le souhait de répartir une collection de règles dans une Spécification de Règles vers l'élément de réseau intermédiaire comprenant

i. l'identification de l'élément de réseau intermédiaire vers lequel la collection de règles dans une Spécification de Règles est répartie,

ii. l'identification de l'élément de réseau intermédiaire depuis lequel la collection des au moins une règles dans une Spécification de Règles est répartie, et

iii. l'identification de la collection de la au moins une règle dans une Spécification de Règles.

8. Dispositif selon l'une quelconque des revendications précédentes, comprenant en outre un moyen de récupération de la collection de règles dans une Spécification de Règles depuis l'élément de réseau intermédiaire qui répartit la collection de règles par l'élément de réseau intermédiaire vers lequel la collection de règles est répartie, comprenant

i. un moyen pour établir un canal de communications entre l'élément de réseau intermédiaire vers lequel la collection de règles est répartie et l'élément de réseau intermédiaire depuis lequel la collection de règles est répartie,

ii. un moyen pour mettre à disposition l'identification de la collection de règles qui est répartie par le biais du canal de communications par l'intermédiaire d'un élément de réseau vers lequel la collection de règles est répartie, et

iii. un moyen pour transmettre la collection de règles qui est répartie par le biais du canal de communications par l'élément de réseau intermédiaire depuis lequel la collection de règles est répartie.

9. Dispositif selon l'une quelconque des revendications précédentes, dans lequel lesdits réseaux de communications comprennent un noeud de point d'extrémité, dénommé ci-après noeud client, pour envoyer une requête vers l'autre noeud de point d'extrémité, dénommé ci-après noeud serveur, par le biais d'au moins un élément de réseau intermédiaire, dans lequel le noeud serveur est adapté pour accepter la requête avec une réponse appropriée, dans lequel des réseaux de communications comprennent en outre un moyen pour régler un canal de communications entre le noeud serveur et le noeud client à travers des éléments de réseau intermédiaire, et dans lequel le noeud serveur est adapté pour débuter une transmission de paquets de données au travers du canal de communications vers le noeud client jusqu'à ce que le noeud client envoie une requête, par le biais des éléments de réseau intermédiaire, pour anéantir le canal de communications, et dans lequel le noeud client est adapté pour retransmettre des informations sur des statistiques de transmission vers le noeud serveur.

10. Dispositif selon la revendication 9, comprenant en outre un moyen permettant à l'auteur de la Spécification de Règles de déclencher une règle unique ou une pluralité de règles au niveau d'un élément de réseau intermédiaire sur la base des procédés de commande suivants

i. la règle devant être évaluée lorsque l'élément de réseau intermédiaire a reçu un paquet de requêtes du noeud client vers le noeud serveur,

ii. la règle devant être évaluée lorsque l'élément de réseau intermédiaire a reçu un paquet de réponses du noeud serveur vers le noeud client,

iii. la règle devant être évaluée lorsque l'élément de réseau intermédiaire a reçu un paquet de données contenant des contenus envoyés par le noeud serveur vers le noeud client au travers du canal de communications établi entre le noeud serveur et le noeud client,

iv. la règle devant être évaluée lorsque l'élément de réseau intermédiaire a reçu un paquet de données contenant les statistiques de transmission du noeud client vers le noeud serveur,

v. la règle devant être évaluée lorsque l'élément de réseau intermédiaire a reçu un nombre spécifié de paquets de données contenant un contenu envoyé par le noeud serveur vers le noeud client au travers du canal de communications établi entre le noeud serveur et le noeud client, et

vi. la règle devant être évaluée lorsque l'élément de réseau intermédiaire a reçu un paquet de données contenant un contenu envoyé par le noeud serveur vers le noeud client au travers du canal de communications établi entre le noeud serveur et le noeud client après qu'un temporisateur récurrent d'une valeur de temporisateur spécifié se soit écoulé.

11. Dispositif selon l'une quelconque des revendications précédentes comprenant un moyen de commande pour utiliser un groupe de paramètres dans la Spécification de Règles pour commander au moins un contenu ou des cessions de distribution de contenu pour obtenir une indépendance de dispositif dans la distribution dudit contenu, comprenant

i. le groupe de paramètres de Préférences d'utilisateur comprenant les préférences de l'utilisateur humain consommant le contenu,

ii. le groupe de paramètres de Possibilités d'Agent comprenant les possibilités de l'agent logiciel utilisé par l'utilisateur humain pour récupérer le contenu,

iii. le groupe de paramètres de Possibilités de Dispositif comprenant les possibilités du matériel utilisé par l'utilisateur humain pour récupérer le contenu, et

iv. le groupe de paramètres d'environnement Naturel comprenant les informations concernant l'environnement dans lequel l'utilisateur humain récupère le contenu.

12. Dispositif selon la revendication 13, dans lequel des paramètres de Préférences d'utilisateur comprennent

i. les préférences de l'utilisateur humain sur le procédé de récupération du contenu,

ii. les préférences de l'utilisateur humain sur le langage utilisé pour récupérer le contenu,

iii. les préférences de l'utilisateur humain sur la présentation du contenu récupéré,

iv. le groupe d'âge de l'utilisateur humain récupérant le contenu,

v. le sexe de l'utilisateur humain récupérant le contenu, et

vi. l'état de l'emploi de l'utilisateur humain récupérant le contenu.

13. Dispositif selon la revendication 11, dans lequel le groupe de paramètres de Possibilités d'Agent comprend

i. le type d'agent logiciel utilisé par l'utilisateur humain pour récupérer le contenu,

ii. les formats de contenu supportés par l'agent logiciel utilisé par l'utilisateur humain pour récupérer le contenu,

iii. les langages de contenu supportés par l'agent logiciel utilisé par l'utilisateur humain pour récupérer le contenu, et

iv. les protocoles de transmission supportés par l'agent logiciel utilisé par l'utilisateur humain pour récupérer le contenu.

14. Dispositif selon la revendication 11, dans lequel le groupe de paramètres de Possibilités de Dispositif comprend

i. le type de matériel utilisé par l'utilisateur humain pour récupérer le contenu

ii. la vitesse de processeur et la famille de processeurs du matériel utilisé par l'utilisateur humain pour récupérer le contenu,

iii. la capacité de mémoire de la mémoire physique de la mémoire secondaire du matériel utilisé par l'utilisateur humain pour récupérer le contenu,

iv. la profondeur d'affichage et la résolution du matériel utilisé par l'utilisateur humain pour récupérer le contenu, et

v. le système d'exploitation fonctionnant sur le matériel utilisé par l'utilisateur humain pour récupérer le contenu.

15. Dispositif selon la revendication 11, dans lequel le groupe de paramètres d'Environnement Naturel comprend

i. les informations de l'emplacement d'où l'utilisateur humain récupère le contenu,

ii. les informations sur la mobilité de l'utilisateur humain récupérant le contenu, et

iii. les informations des conditions d'éclairement dans lesquelles l'utilisateur humain récupère le contenu.

16. Dispositif selon l'une quelconque des revendications 11 à 14, dans lequel le au moins un progiciel spécial est capable d'interpréter et d'évaluer ladite Spécification de Règles.

17. Procédé de structure de commande de réseau pour commander des ressources au niveau d'un élément de réseau intermédiaire connectant deux réseaux de communications ou plus comprenant les étapes consistant à :

a) fournir une fonctionnalité de passerelle par un module de passerelle,

b) réaliser une décision de commande de ressources par un module de moteur de règles sur la base de règles spécifiées, dans lesquelles les règles sont spécifiées dans un format de spécification de règles dénommé ci-après une Spécification de Règles,

c) offrir une fonctionnalité spécialisée au module de moteur de règles par au moins un progiciel spécial ajouté sur le module de moteur de règles,

d) injecter dans, ou retirer une spécification de règles du module de moteur de règles par un module d'injection de règles, et

e) une répartition de ladite Spécification de Règles vers au moins un élément de réseau intermédiaire comprenant les étapes consistant en

i. une répartition d'indications dans la Spécification de Règles pour indiquer qu'une partie ou la totalité de la Spécification de Règles va être répartie,

ii. une répartition d'une signature intégrée dans des paquets de données pour annoncer les possibilités des éléments de réseau intermédiaire traversés par les paquets de données,

iii. une analyse de la Spécification de Règles pour déterminer si une partie ou la totalité de la spécification de règles spécifiées est répartie,

iv. identifier l'élément de réseau cible pour répartir une partie ou la totalité d'une Spécification de Règles,

v. une répartition d'une signalisation intégrée dans des paquets de données pour informer l'élément de réseau cible de la répartition d'une partie ou de la totalité d'une Spécification de Règles,

vi. une répartition de la partie ou de la totalité de la Spécification de Règles répartie vers l'élément de réseau cible depuis l'élément de réseau intermédiaire qui répartit la partie ou la totalité de la Spécification de Règles.

18. Procédé selon la revendication 17 comprenant en outre une étape d'extraction de la signature des éléments de réseau intermédiaire intégrée dans au moins un paquet de données, comprenant les étapes consistant à

i. vérifier s'il existe des signatures intégrées dans les paquets de données,

ii. vérifier s'il existe une signature dans un format de données prédéterminées qui a été mémorisé auparavant ayant les mêmes points de départ et d'arrivée et un protocole de transmission,

iii. attribuer un nouveau format de données lorsqu'il n'existe pas de format de données qui a été mémorisé auparavant ayant les mêmes points de départ et d'arrivée et un protocole de transmission,

iv. purger des données mémorisées dans le format de données qui existait auparavant ayant les mêmes points de départ, d'arrivée et un protocole de transmission,

v. préparer une structure de données dernier entré-premier sorti vide,

vi. extraire chaque signature intégrée dans le paquet de données et la pousser vers la structure de données dernier entré-premier sorti,

vii. retirer chaque élément dans la structure de données dernier entré-premier sorti et l'enregistrer dans le format de données prédéterminé, et

viii. enregistrer le nombre de signatures intégrées extraites dans le format de données prédéterminé.

19. Procédé selon l'une des revendications 17 ou 18, comprenant en outre une étape d'analyse d'une Spécification de Règles pour déterminer si une partie ou la totalité de la Spécification de Règles va être répartie, comprenant les étapes consistant à

i. vérifier chaque règle dans la Spécification de Règles pour une validité syntaxique,

ii. rejeter la règle s'il existe des erreurs syntaxiques,

iii. vérifier la règle pour une indication de répartition,

iv. évaluer la règle localement s'il n'existe pas d'indication de répartition,

v. déterminer l'élément de réseau intermédiaire vers lequel répartir la règle,

vi. évaluer la règle localement si aucun élément de réseau intermédiaire distant vers lequel répartir la règle ne peut être trouvé,

vii. vérifier si l'élément de réseau intermédiaire contient le progiciel spécial ou les progiciels spéciaux nécessaires dans la règle,

viii. évaluer la règle localement si l'élément de réseau intermédiaire distant n'a pas le progiciel spécial ou les progiciels spéciaux requis, et

ix. répartir la règle vers l'élément de réseau intermédiaire distant.

20. Procédé selon la revendication 17, comprenant en outre un procédé de détermination de l'élément de réseau intermédiaire distant vers lequel une règle va être répartie en fonction d'une indication de répartition prédéterminée, comprenant les étapes consistant à

i. situer une signature dans un format de données prédéterminé avec le point de départ, le point d'arrivée et le protocole de transmission correspondants,

ii. ne déclarer aucun élément de réseau intermédiaire distant approprié si aucun format de données prédéterminé ne peut être situé,

iii. régler une variable temporaire par rapport au nombre spécifié d'intermédiaires vers ou à partir du point d'extrémité spécifié dans l'indication de répartition donnée,

iv. régler la variable temporaire par rapport à la valeur du nombre d'intermédiaires telle que donnée dans le format de données prédéterminé situé si le nombre spécifié d'éléments de réseau intermédiaire vers ou depuis le point d'extrémité spécifié dans l'indication de répartition donnée est plus grand que le nombre d'éléments de réseau intermédiaire vers ou depuis le point d'arrivé spécifié dans l'indication de répartition donnée,

v. si l'indication de répartition spécifiée est composée de la spécification du point d'arrivée et de la spécification du nombre d'éléments de réseau intermédiaire vers le point d'arrivée spécifié, régler la variable temporaire sur une valeur égale au nombre d'éléments de réseau intermédiaire donné dans le format de données prédéterminé situé moins la valeur d'origine dans la variable temporaire,

vi. si l'indication de répartition spécifiée est composée de la spécification du point d'arrivée et de la spécification du nombre d'éléments de réseau intermédiaire depuis le point d'extrémité supérieur, régler la variable temporaire sur une valeur égale à la valeur d'origine dans la variable temporaire moins 1,

vii. déclarer l'élément de réseau intermédiaire distant comme étant l'élément de réseau spécifié dans une signature mémorisée dans le format de données prédéterminé situé où la signature a un indice dans l'ensemble de signatures dans le format de données prédéterminé situé égal à la valeur mémorisée dans la variable temporaire, si un tel indice existe, et

viii. déclarer aucun élément de réseau intermédiaire distant approprié si l'indice égal à la valeur mémorisée dans la variante temporaire n'existe pas dans l'ensemble de signatures dans le format de données prédéterminé situé.

21. Réseau de communications comprenant un dispositif de structure de commande de réseau selon l'une quelconque des revendications 1 à 16 pour commander des ressources au niveau d'un élément de réseau intermédiaire connectant deux réseaux de communications ou plus.

Fig.1

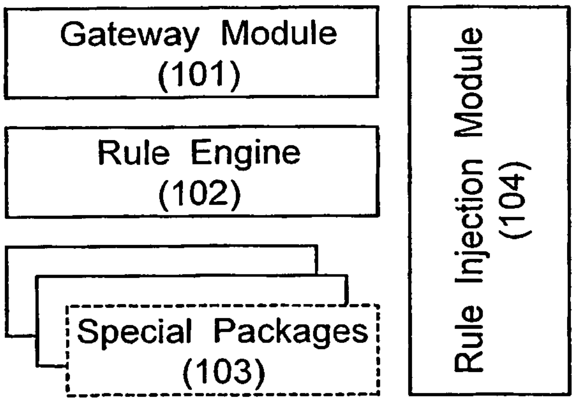

Gateway Module
(101)
Rule Engine
(102)
Special Packages
(103)
Rule Injection Module
(104)

Fig.2

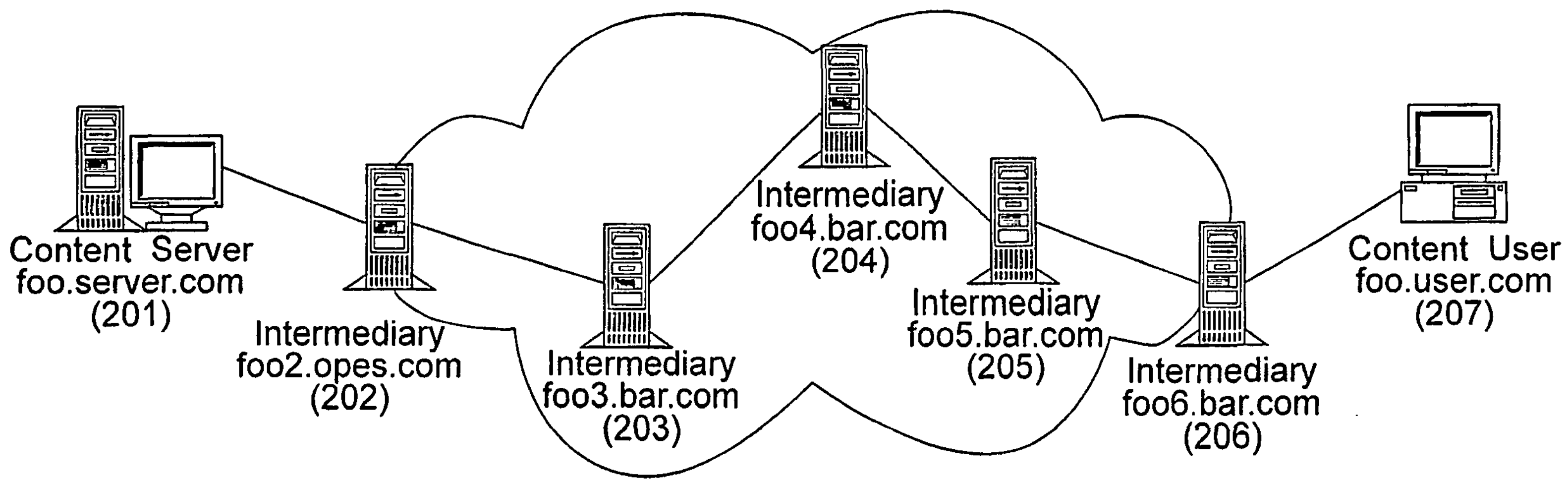

Content Server
foo.server.com
(201)
Intermediary
foo2.opes.com
(202)
Intermediary
foo3.bar.com
(203)
Intermediary
foo4.bar.com
(204)
Intermediary
foo5.bar.com
(205)
Intermediary
foo6.bar.com
(206)
Content User
foo.user.com
(207)

Fig.3

```
ContentPath = {
        source = “foo.user.com”
        destination = “foo.server.com”
        protocol = ...
        num_nodes = 2
        nodes[0] = {
                hostname = “foo3.bar.com”
                capabilities = ...
        }
  nodes[1] = {
                hostname = “foo2.bar.com”
                capabilities = ...
        }
}

ContentPath = {
        source = “foo.server.com”
        destination = “foo.user.com”
        protocol = ...
        num_nodes = 2
        nodes[0] = {
                hostname = “foo5.bar.com”
                capabilities = ...
        }
        nodes[1] = {
                hostname = “foo6.bar.com”
        capabilities = ...
        }
}
```

Fig.4

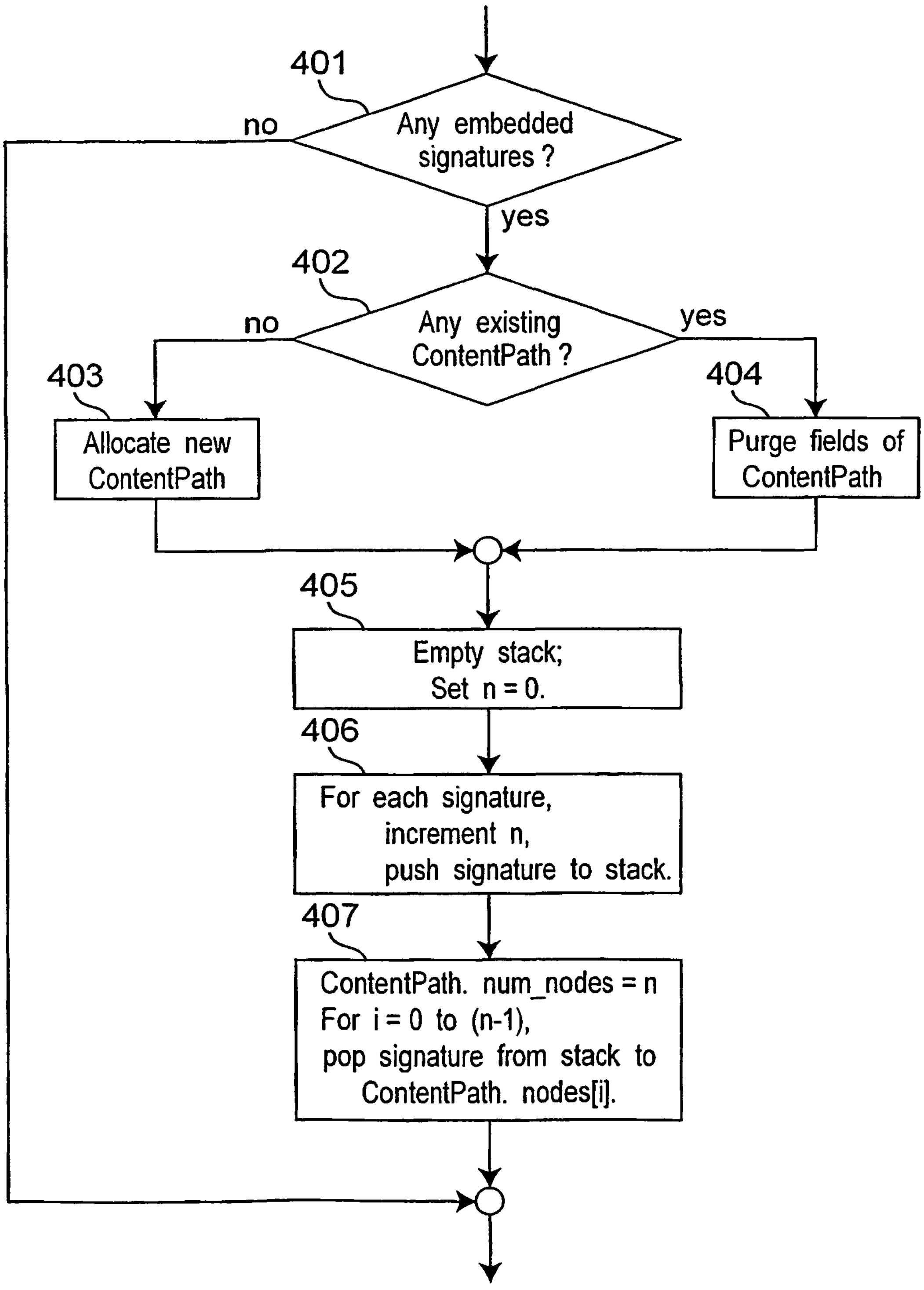

401
Any embedded signatures ?
no
yes
402
Any existing ContentPath ?
no
yes
403
Allocate new ContentPath
404
Purge fields of ContentPath
405
Empty stack; Set n = 0.
406
For each signature, increment n, push signature to stack.
407
ContentPath. num_nodes = n For i = 0 to (n-1), pop signature from stack to ContentPath. nodes[i].

Fig.5

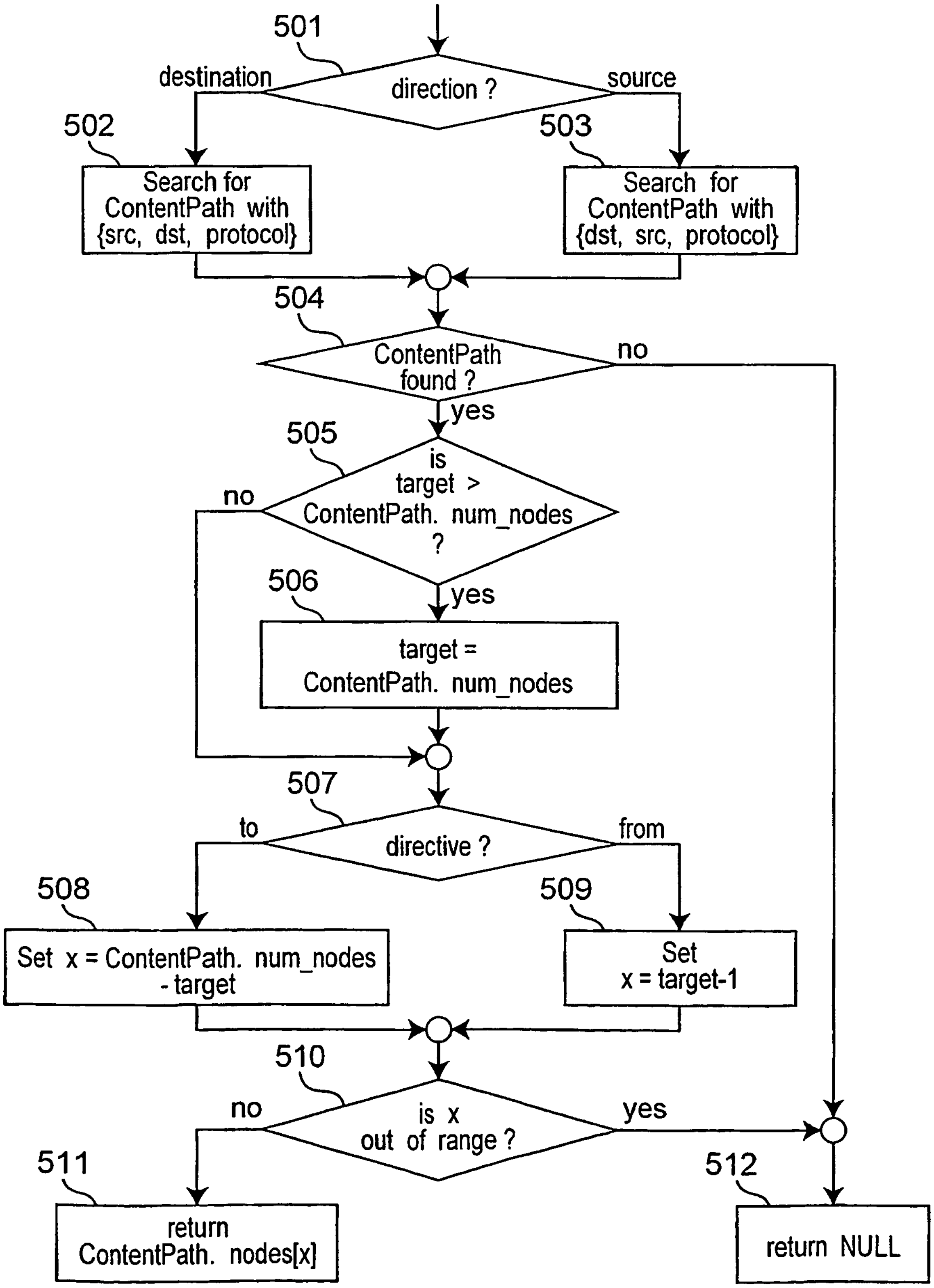

501
direction ?
destination
source
502
Search for ContentPath with {src, dst, protocol}
503
Search for ContentPath with {dst, src, protocol}
504
ContentPath found ?
no
yes
505
is target > ContentPath. num_nodes ?
no
yes
506
target = ContentPath. num_nodes
507
directive ?
to
from
508
Set x = ContentPath. num_nodes - target
509
Set x = target-1
510
is x out of range ?
no
yes
511
return ContentPath. nodes[x]
512
return NULL

Fig.6

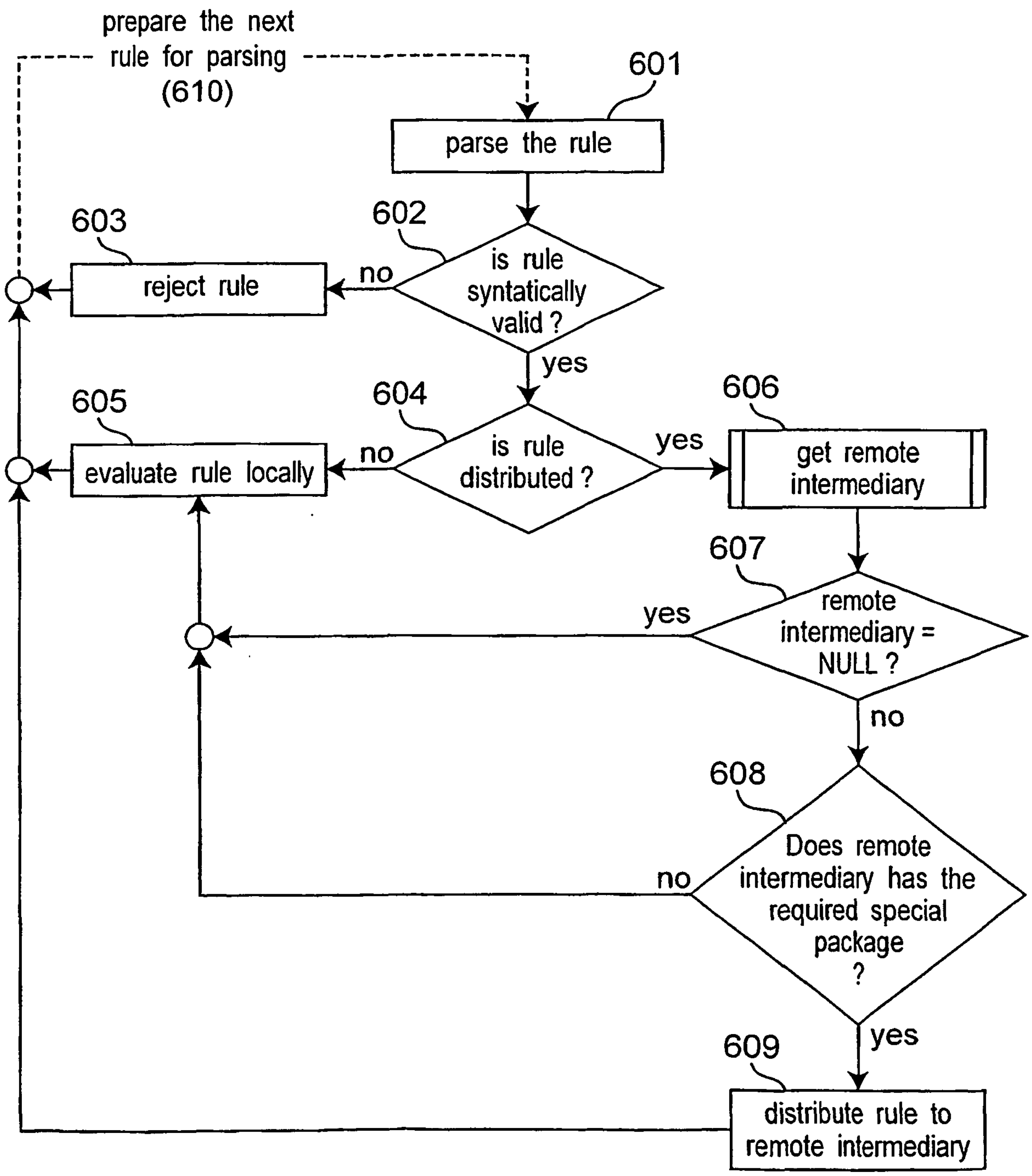
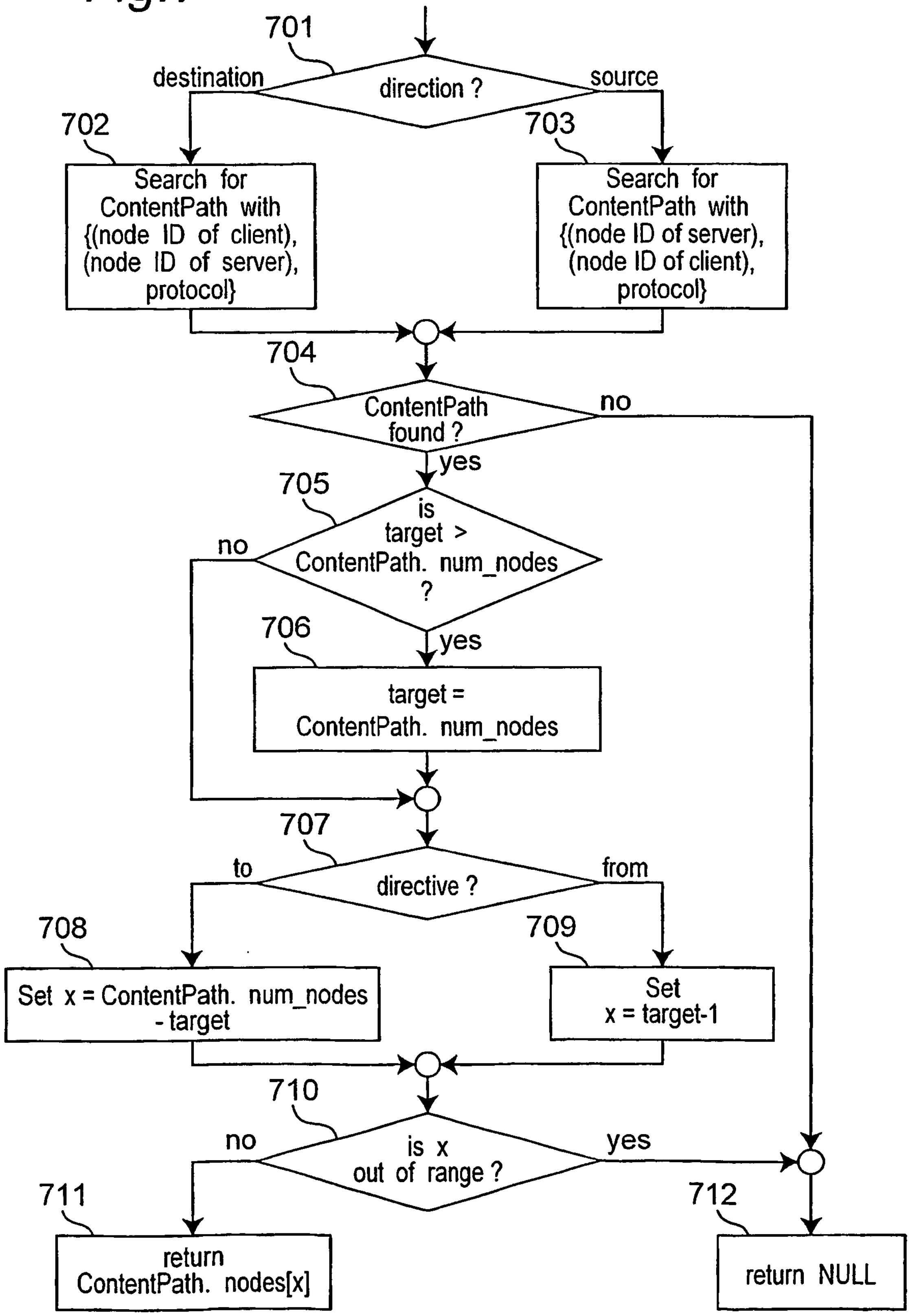

prepare the next rule for parsing (610)
601
parse the rule
602
is rule syntatically valid ?
no
603
reject rule
yes
604
is rule distributed ?
no
605
evaluate rule locally
yes
606
get remote intermediary
607
remote intermediary = NULL ?
yes
no
608
Does remote intermediary has the required special package ?
no
yes
609
distribute rule to remote intermediary

Fig.7


701
direction ?
destination
source
702
Search for ContentPath with {(node ID of client), (node ID of server), protocol}
703
Search for ContentPath with {(node ID of server), (node ID of client), protocol}
704
ContentPath found ?
no
yes
705
is target > ContentPath. num_nodes ?
no
yes
706
target = ContentPath. num_nodes
707
directive ?
to
from
708
Set x = ContentPath. num_nodes - target
709
Set x = target-1
710
is x out of range ?
no
yes
711
return ContentPath. nodes[x]
712
return NULL